



T-404-LOKA LOKAVERKEFNI

MAREL

CLOUD ANALYTICS

---

# Development Manual

---

*Students:*

Gísli Rafn Guðmundsson  
Gunnar Páll Gunnarsson  
Jón Reginbald Ívarsson

*Teacher:*

Hallgrímur Arnalds

*Instructor:*

Elín Elísabet Torfadóttir

*Examiner:*

Haukur Kristinsson

May 13, 2016

# Contents

<b>1</b>	<b>Preface</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Maintenance . . . . .	3
1.3	Version . . . . .	3
1.3.1	History . . . . .	3
<b>2</b>	<b>Visual Studio 2013</b>	<b>4</b>
2.1	Install . . . . .	4
2.2	Update . . . . .	4
2.3	Extensions . . . . .	4
2.3.1	.NET Framework . . . . .	4
2.3.2	Azure SDK . . . . .	4
2.3.3	PowerShell Tools . . . . .	4
2.4	Version Control . . . . .	5
2.5	Project Solution . . . . .	9
<b>3</b>	<b>Solution Structure</b>	<b>10</b>
3.1	Scripts . . . . .	10
3.2	IoT Hub . . . . .	10
3.3	SourceControl . . . . .	10
3.4	SQL Database . . . . .	10
3.5	Stream Analytics . . . . .	10
3.6	Web App . . . . .	10
<b>4</b>	<b>Web Application</b>	<b>11</b>
4.1	Node.js and npm . . . . .	11
4.1.1	Install . . . . .	11
4.1.2	Run . . . . .	11
4.2	Git . . . . .	11
4.2.1	Install . . . . .	11
4.3	Bower . . . . .	11
4.3.1	Prerequisite . . . . .	11
4.3.2	Install . . . . .	11
4.3.3	Run . . . . .	12
4.4	Gulp . . . . .	12
4.4.1	Install . . . . .	12
4.4.2	Run . . . . .	12
4.4.3	Configuration . . . . .	12
4.5	Karma & Jasmine . . . . .	12
4.5.1	Prerequisite . . . . .	12
4.5.2	Install . . . . .	12
4.5.3	Run . . . . .	13
4.5.4	Configuration . . . . .	13
4.6	Script . . . . .	13
4.7	Further reading . . . . .	13

<b>5</b>	<b>Azure</b>	<b>14</b>
5.1	Portal . . . . .	14
5.2	Further reading . . . . .	14
<b>6</b>	<b>Testing</b>	<b>15</b>
6.1	Testing the API back-end & the CloudLogger class . . . . .	15
6.2	Testing the Angular front-end . . . . .	15
6.3	End-to-end Testing . . . . .	15
<b>7</b>	<b>Scrum</b>	<b>16</b>
<b>8</b>	<b>Continuous Integration &amp; Deployment</b>	<b>16</b>
<b>9</b>	<b>Useful Tools</b>	<b>17</b>
9.1	ISO Mount . . . . .	17
9.2	SSH client . . . . .	17
9.3	Microsoft Azure Storage Explorer . . . . .	17
9.4	Device Explorer . . . . .	17

# 1 Preface

*Cloud Analytics* is an *Innova* monitoring and analytics solution. It utilizes a plethora of different tools and prerequisites that are needed for the development process.

## 1.1 Purpose

The purpose of this document is to give a step-by-step guide for developers to setup a development environment for the Cloud Analytics project and start developing.

## 1.2 Maintenance

This document is maintained by *Team Pretzel*.

If you have any issues or problems with this manual, please contact:

jon.reginbald@marel.com

## 1.3 Version

Current document version is 1.0.0 as of 13.05.2016.

### 1.3.1 History

Date	Version	Comments	Author
13.05.2016	1.0.0	Initial version.	JRI

## 2 Visual Studio 2013

*Visual Studio* is an integrated development environment from Microsoft and is the heart of our development process.

### 2.1 Install

Install Visual Studio using the *Marel* provided .iso file.

### 2.2 Update

Install the Microsoft Visual Studio 2013 Update 5 from this link:

<https://www.microsoft.com/en-us/download/details.aspx?id=48129>

### 2.3 Extensions

These extensions are required to interface Visual Studio with the *Azure* cloud platform, compile the back-end and run PowerShell modules.

#### 2.3.1 .NET Framework

Install .NET 4.6.1 for Visual Studio 2013 using this link:

<http://getdotnet.azurewebsites.net/target-dotnet-platforms.html>

#### 2.3.2 Azure SDK

Install the Azure SDK for .NET Visual Studio 2013 using this link:

<https://azure.microsoft.com/en-us/downloads/>

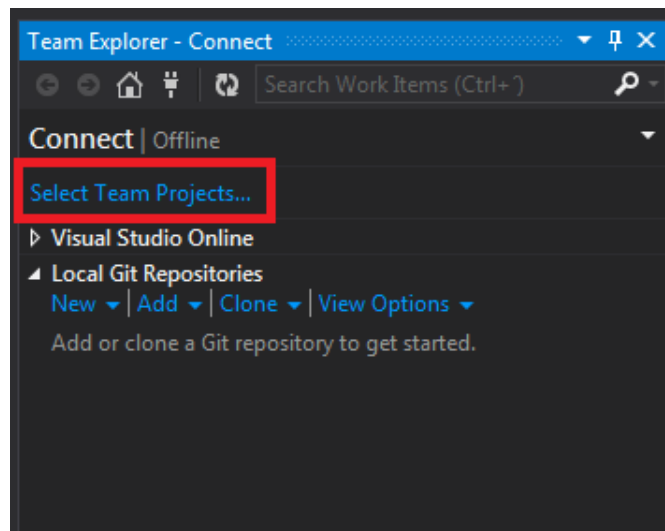
#### 2.3.3 PowerShell Tools

Install a set of tools for developing and debugging PowerShell scripts and modules in Visual Studio.

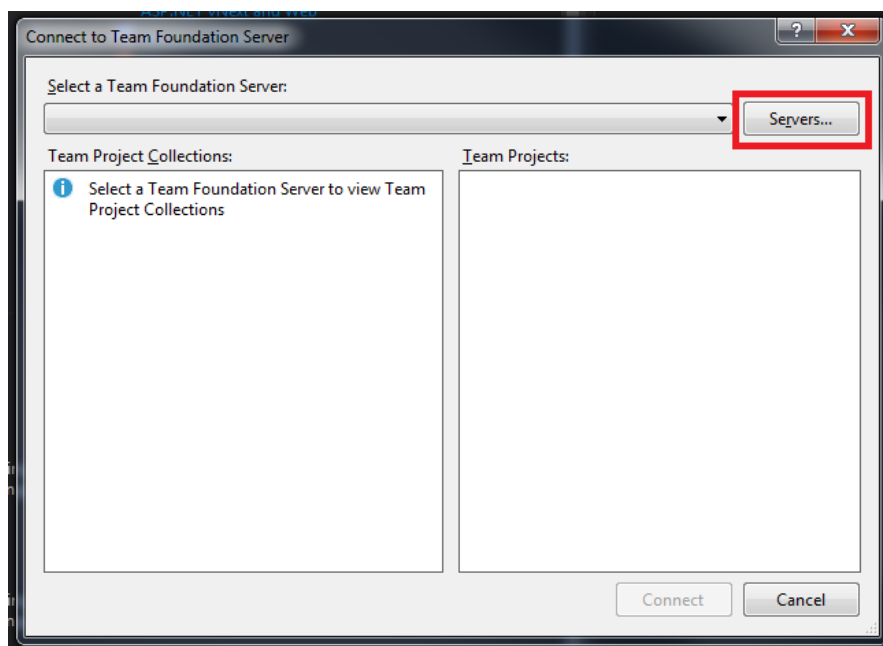
<https://visualstudiogallery.msdn.microsoft.com/f65f845b-9430-4f72-a182-ae2a7b8999d7>

## 2.4 Version Control

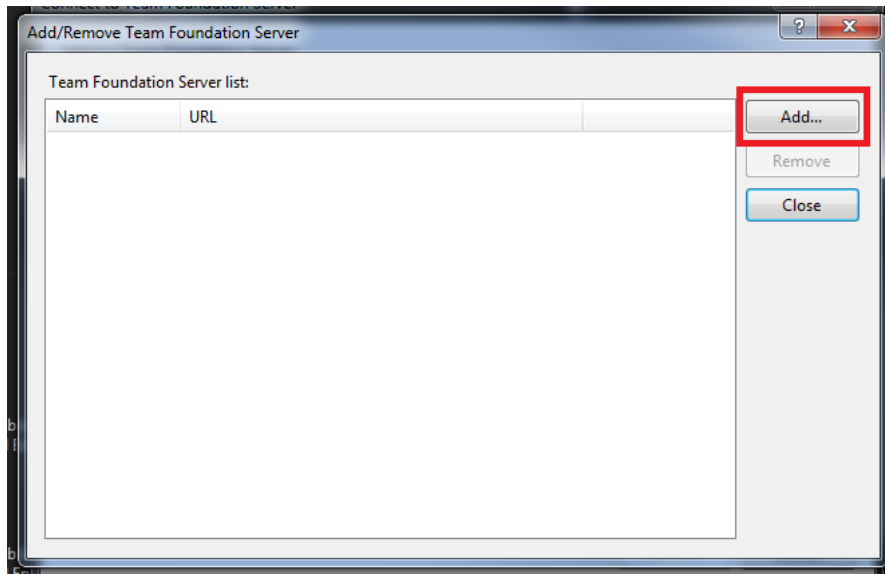
Take the following steps to connect Visual Studio to Visual Studio Team Services. Start by opening *Team Explorer* in Visual Studio.



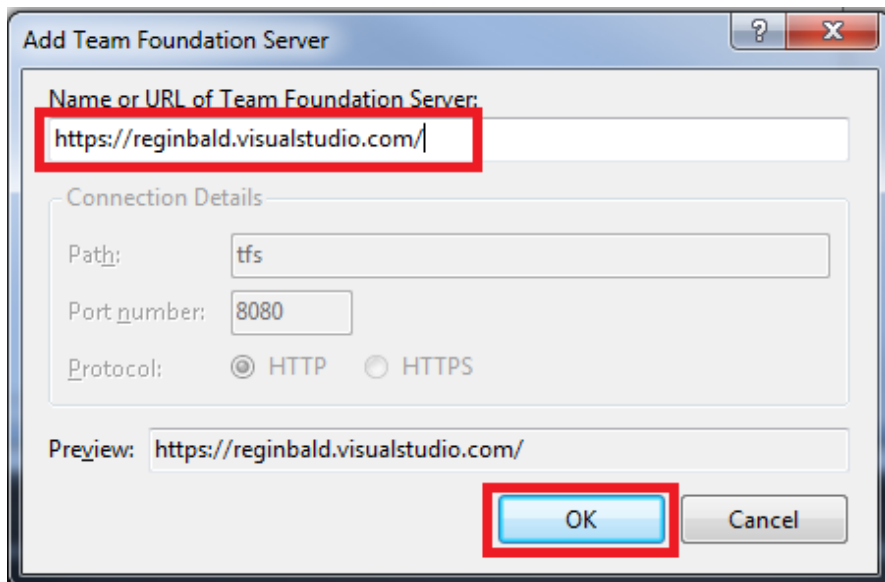
Open *Select Team Projects...* dialog



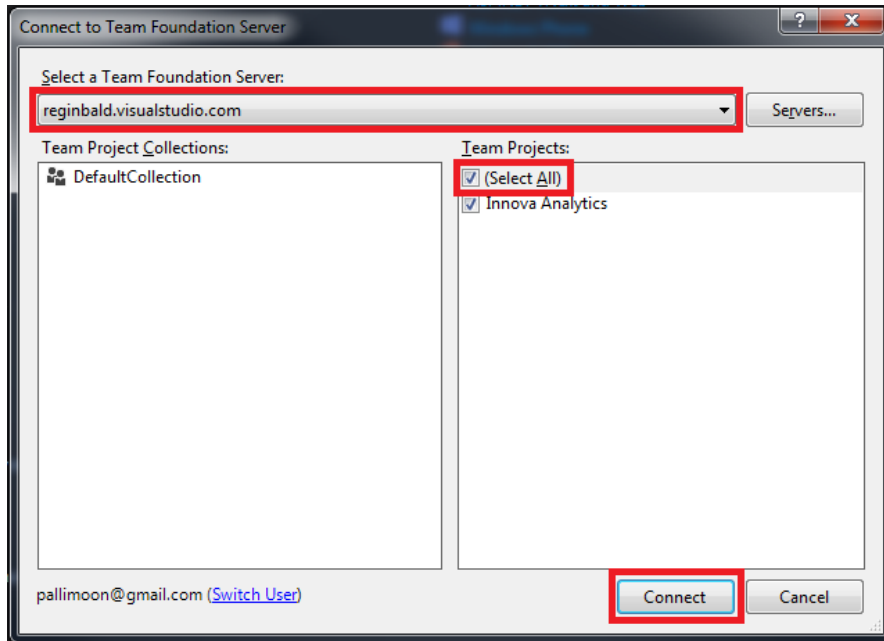
Open the *Servers...* dialog



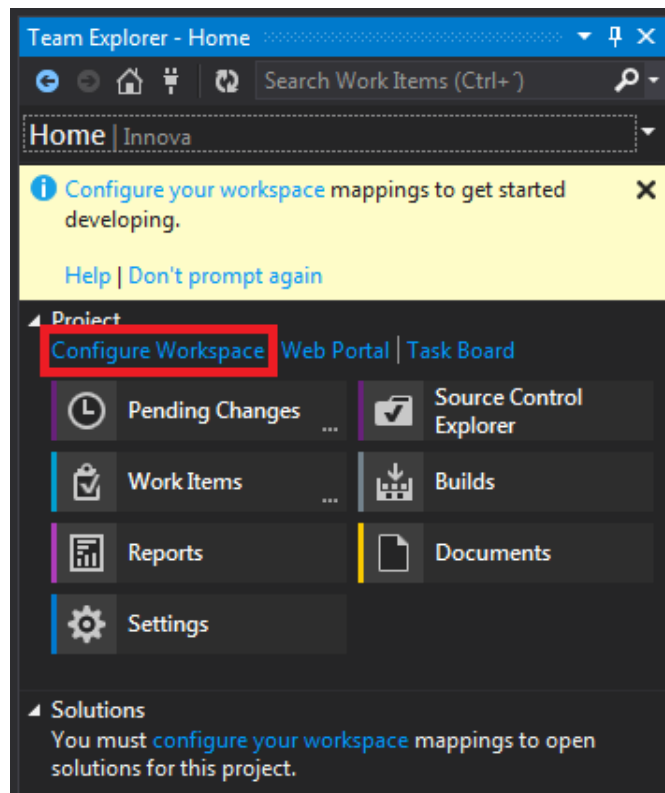
Open the *Add...* dialog



Insert the URL *https://reginbald.visualstudio.com/* into the first input field and press *OK*

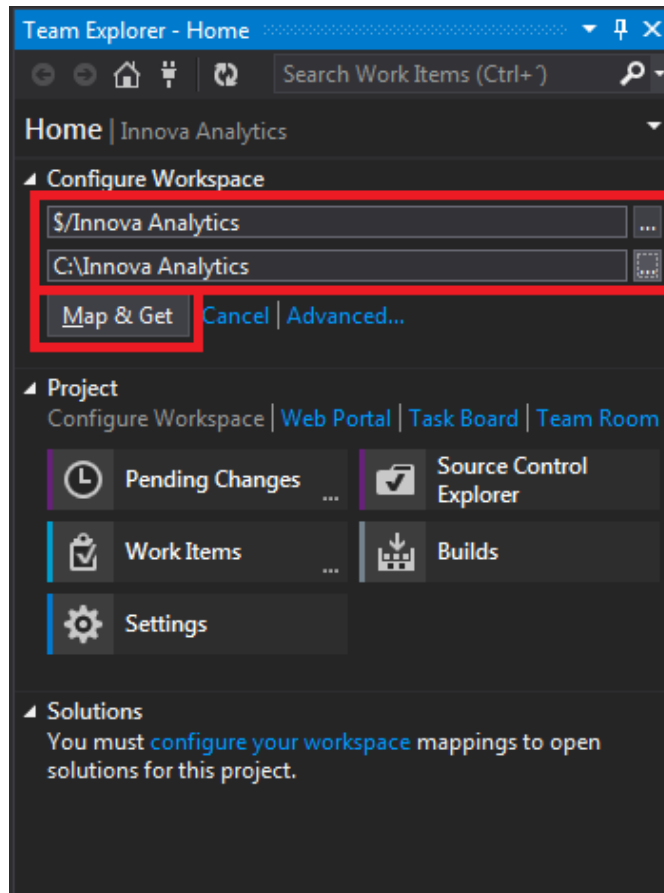


Select the new server from the drop-down menu and *Select All* team projects

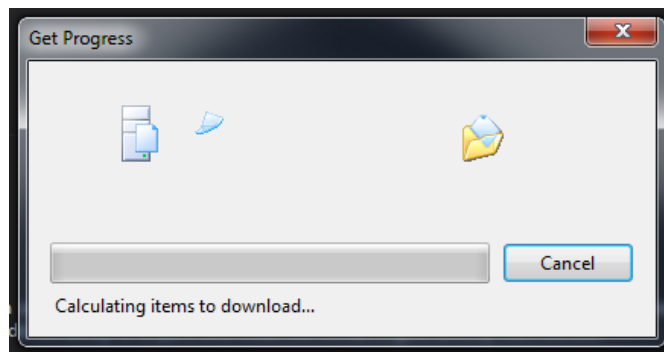


Open the *Configure Workspace* dialog





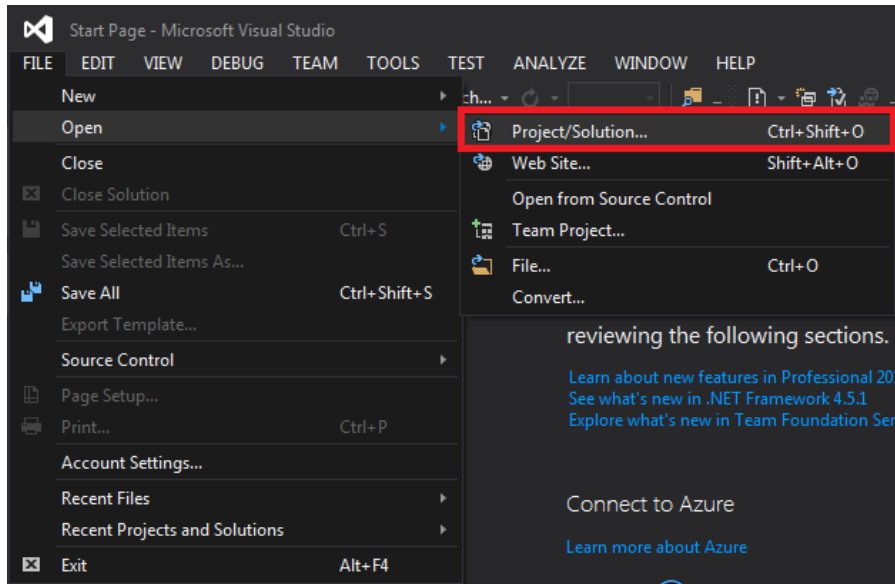
Choose where to locate the workspace (the lower field), then click *Map & Get*



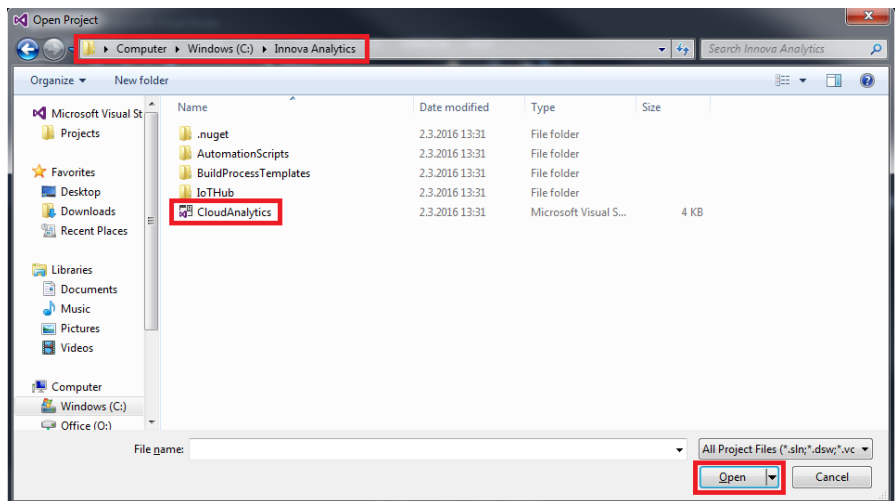
Wait for the download to finish

## 2.5 Project Solution

Open the Visual Studio solution using the following steps.



*File → Open → Project/Solution...*



Navigate to the *Innova Analytics* folder, *C:\Innova Analytics\*, select the *Visual Studio* solution and press *Open*

## 3 Solution Structure

This section will review the solution structure in details. Each sub-section corresponds to a folder in the solution.

### 3.1 Scripts

The *Scripts* folder contains a PowerShell Project named *AutomationScripts* and a folder named *End2EndTests*.

- **AutomationScripts** project contains scripts that are used to provision the Azure cloud services and deploy a web application. They are covered in more detail in the *Administration Manual*.
- **End2EndTests** folder contains a Node.js application to test and verify a newly provisioned system.

### 3.2 IoT Hub

There are three projects in the *IoT Hub* folder.

- **CloudLogger** contains the code for communication between Innova systems and IoT Hub.
- **CloudLoggerTest** contains unit tests for the *CloudLogger* project.
- **SimpleLogDevice** contains code for a command line application that sends random logs to IoT Hub.

### 3.3 SourceControl

The *SourceControl* directory contains a configuration file for the source control.

### 3.4 SQL Database

This folder contains SQL files for creating, inserting and deleting database tables.

### 3.5 Stream Analytics

This folder contains SQL files for configuring Stream Analytics jobs.

### 3.6 Web App

Contains four projects.

- **InnovaRegistry** contains code for the API end point and the angular web app.
- **InnovaRegistry.Models** contains all the models used by the API back-end.
- **InnovaRegistry.Services** contains the business logic for the API back-end.
- **InnovaRegistry.Tests** contains unit tests for the API back-end.

## 4 Web Application

The Web Application is built with AngularJS. Angular is an open source JavaScript MVW framework maintained by Google that enables developers to build structured, easily testable, and maintainable front-end applications. Development for the web application is done in the *App* folder in the *InnovaRegistry* project. **All the commands below should be executed in this directory.**

### 4.1 Node.js and npm

*Node.js* is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. npm is Node's packages manager. It runs through the command line and manages dependencies for the application.

#### 4.1.1 Install

Install the current version of Node.js and npm using this link: <https://nodejs.org/en/>

#### 4.1.2 Run

Run the following command to install the necessary npm dependencies:

```
$ npm install
```

### 4.2 Git

*Git* is a free and open source distributed version control system. For this project, Git is mainly used by Bower to download front-end dependencies. Additionally we recommend using the Git Bash command line tool included to start Gulp.

#### 4.2.1 Install

Install Git using this link: <https://git-scm.com/download/win>

### 4.3 Bower

*Bower* is a package manager for the web. It depends on Node.js and npm and works with Git and GitHub repositories.

#### 4.3.1 Prerequisite

- Git must be installed on the system.

#### 4.3.2 Install

Install Bower using this command using your preferred command line interface:

```
$ npm install -g bower
```

### 4.3.3 Run

Run the following command to install Bower dependencies:

```
$ bower install
```

## 4.4 Gulp

Gulp is a task runner that helps developers automate painful or time-consuming tasks in their development workflow.

### 4.4.1 Install

Install Gulp using this command in Git Bash:

```
$ npm install -g gulp
```

### 4.4.2 Run

Run the following command to build the web app:

```
$ gulp
```

Run the following command to build the web app for production:

```
$ gulp --production
```

### 4.4.3 Configuration

All Gulp tasks can be found in the *gulp* folder.

## 4.5 Karma & Jasmine

*Karma* is a JavaScript test runner and *Jasmine* is an open source testing framework for JavaScript.

### 4.5.1 Prerequisite

- Chrome web browser must be installed on the system.

### 4.5.2 Install

Run the following command to install Karma and Jasmine:

```
$ npm install -g jasmine
$ npm install -g jasmine-core
$ npm install -g karma
$ npm install -g karma-chrome-launcher
$ npm install -g karma-jasmine
$ npm install -g karma-coverage
$ npm install -g karma-ng-html2js-preprocessor
$ npm install -g karma-babel-preprocessor
$ npm install -g babel-preset-es2015
```

### 4.5.3 Run

Run the following command to start unit testing the web app.

```
$ karma start
```

### 4.5.4 Configuration

To configure Karma use the *karma.conf.js* configuration file.

## 4.6 Script

Under the *Scripts > Automation Scripts > Misc* folder you will find the PowerShell script *Install.Globally.ps1* which will install all npm global dependencies for the project.

## 4.7 Further reading

- AngularJS documentation: <https://docs.angularjs.org/guide>
- Gulp documentation: <https://github.com/gulpjs/gulp/blob/master/docs/README.md>
- Jasmine documentation: <http://jasmine.github.io/edge/introduction.html>

## 5 Azure

*Microsoft Azure* is a cloud computing platform with a growing collection of integrated cloud services. Azure enables developers to easily build, deploy and manage applications and services through a global network of Microsoft-managed data centers.

### 5.1 Portal

*Azure Portal* is a management solution used to manage various Azure resources. The portal gives developers an end-to-end solution for almost all of the tools they need to develop, deploy and manage their apps. Use this link to access the Azure portal:

<https://portal.azure.com/>

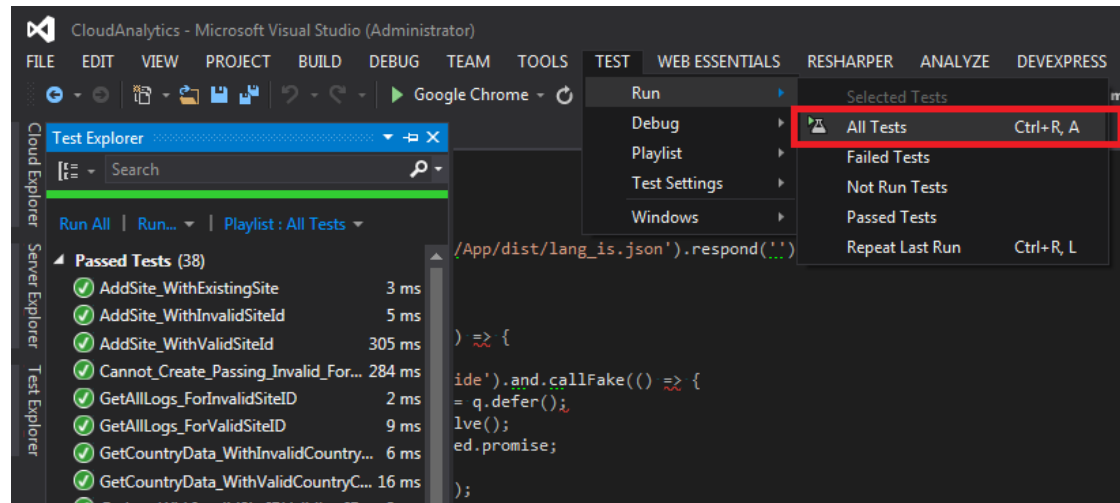
### 5.2 Further reading

- Azure documentation <https://azure.microsoft.com/en-us/documentation/>

## 6 Testing

### 6.1 Testing the API back-end & the CloudLogger class

To run all unit tests navigate to *TEST > Run > All Tests*, in Visual Studio.

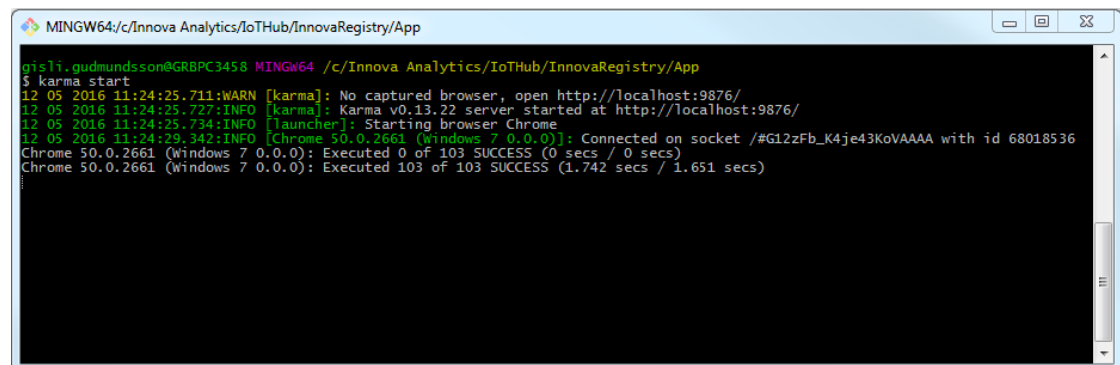


Wait for the download to finish

### 6.2 Testing the Angular front-end

Navigate to the *App* directory in *InnovaRegistry* project using your preferred command line tool. Use the following command to start running the unit tests.

```
$ karma start
```



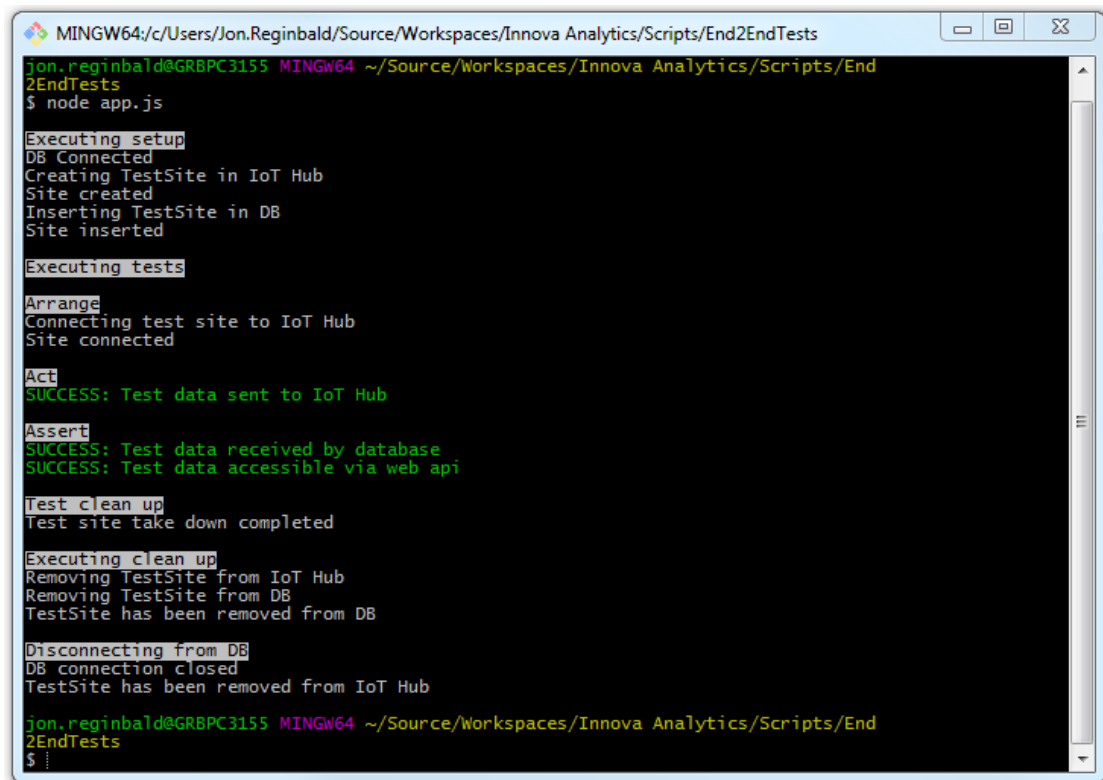
karma start

### 6.3 End-to-end Testing

Navigate to the *End2EndTests* directory in *Scripts* folder using your preferred command line tool. Use the following command to start running the end-to-end tests.

```
$ node app.js
```





```
MINGW64:/c:/Users/Jon.Regimbald/Source/Workspaces/Innova Analytics/Scripts/End2EndTests
jon.regimbald@GRBPC3155 MINGW64 ~/Source/Workspaces/Innova Analytics/Scripts/End2EndTests
$ node app.js

Executing setup
DB Connected
Creating TestSite in IoT Hub
Site created
Inserting TestSite in DB
Site inserted

Executing tests

Arrange
Connecting test site to IoT Hub
Site connected

Act
SUCCESS: Test data sent to IoT Hub

Assert
SUCCESS: Test data received by database
SUCCESS: Test data accessible via web api

Test clean up
Test site take down completed

Executing clean up
Removing TestSite from IoT Hub
Removing TestSite from DB
TestSite has been removed from DB

Disconnecting from DB
DB connection closed
TestSite has been removed from IoT Hub

jon.regimbald@GRBPC3155 MINGW64 ~/Source/Workspaces/Innova Analytics/Scripts/End2EndTests
$
```

node app.js

## 7 Scrum

Backlog and *Kanban* board is accessible through the Visual Studio Team Services website.  
<https://regimbald.visualstudio.com/>

## 8 Continuous Integration & Deployment

All code is checked into *Visual Studio Team Services*, where an on-premise build server builds and tests the project. All successful builds are then deployed to Azure while unsuccessful builds are reported to developers via email.

## 9 Useful Tools

### 9.1 ISO Mount

A tool that allows you to mount optical disc images.

*<http://wincdemu.sysprogs.org/>*

### 9.2 SSH client

An SSH client is used to connect to a remote computer.

*<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>*

### 9.3 Microsoft Azure Storage Explorer

*Microsoft Azure Storage Explorer* is a standalone app from Microsoft that allows you to easily work with Azure Storage data.

Download it from here *<http://storageexplorer.com/>*

### 9.4 Device Explorer

A great tool for creating devices and communicating with an IoT Hub.

Instructions and download at:

*<https://www.github.com/Azure/azure-iot-sdks/tree/master/tools/DeviceExplorer>*