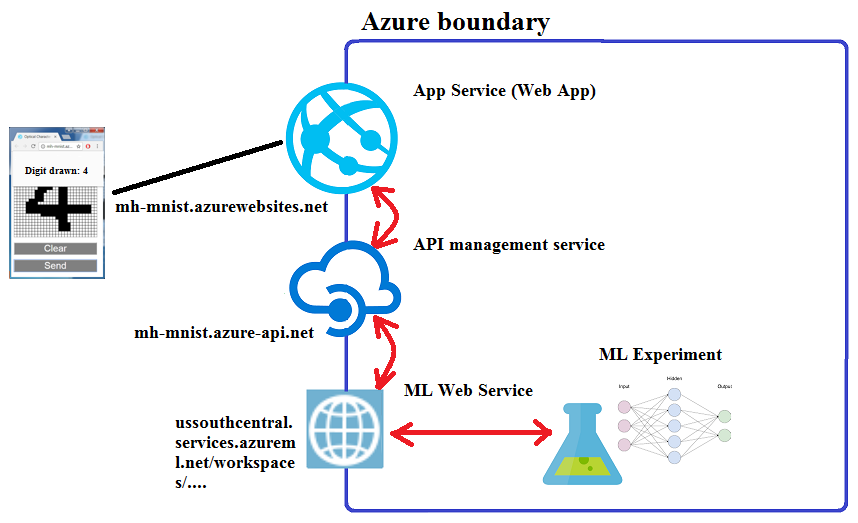
# Outline

Steps to build/deploy handwritten digit recognition application:

* From Session 25, deploy trained MNIST as a ML Web Service
* Create a publicly accessible API (which will connect to the ML Web Service)
* Create a local web handwritten digit recognition application which calls the API to predict digit drawn
* Deploy local digit application as an Azure App Service Web App – to make publicly available



Prerequisite

* Completed worksheet 25

Reference

<https://github.com/Azure-Readiness/hol-azure-machine-learning/blob/master/010-lab-cs-ocr.md>

# Deploy MNIST Experiment As a ML Web Service

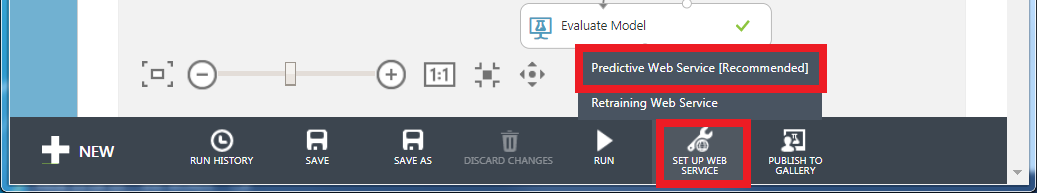
In this section after logging in you will:

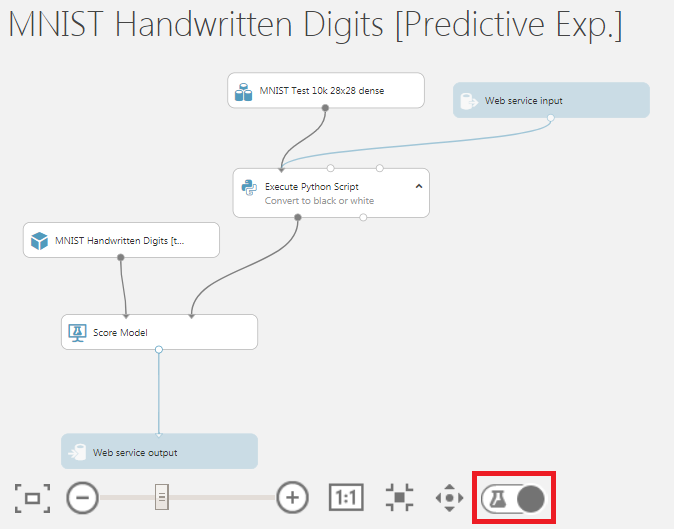
* Deploy MNIST experiment as a ML Web Service

1. Open browse at

<https://studio.azureml.net/>

1. Select ‘Sign In’ and sign in to Azure Machine Learning Studio with your Azure account
2. Select ‘MNIST Handwritten Digits’ experiment
3. Select ‘Run’ to ensure all the modules are up to date.
4. Select ‘SET UP WEB SERVICE’, ‘Predictive Web Service’

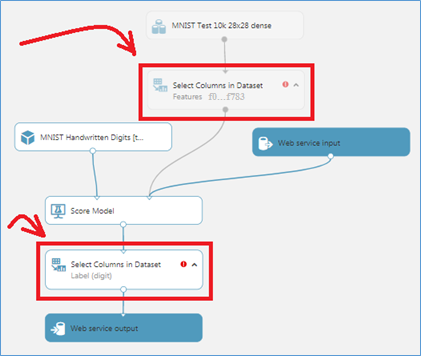




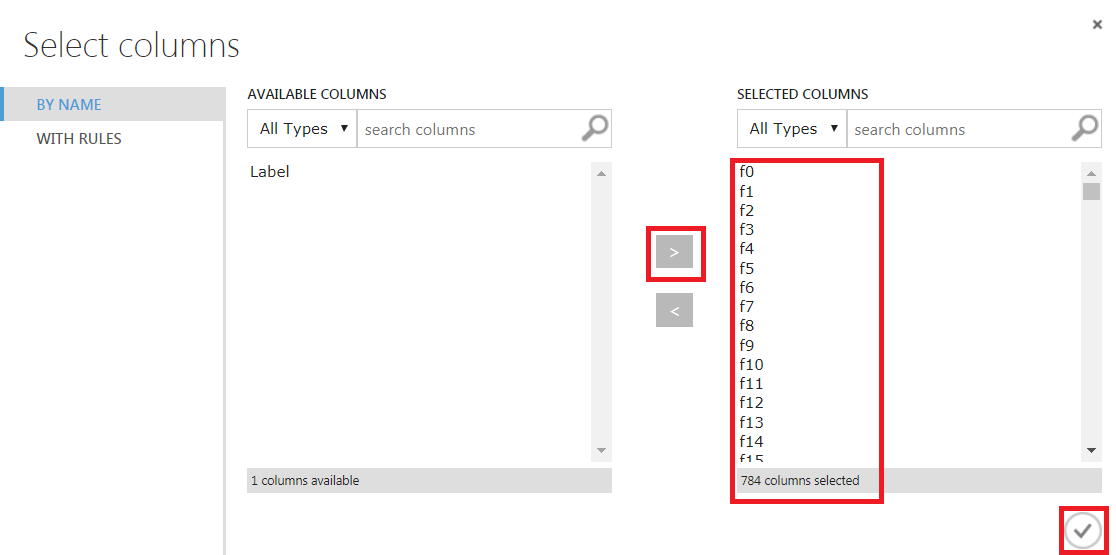
A new ‘Predictive experiment’ tab should appear, by default ‘Web service input’ is added as shown at the top but is in wrong place for this experiment.

1. Click on the beaker icon near the bottom so ‘Web service input’ is not dimmed out
2. Remove ‘Execute Python Script’
3. From the pallet select ‘Data Transformation’, ‘Manipulation’ then drag and drop two ‘Select Columns in Dataset’ module onto canvas
4. Connect modules as shown below and comment top ‘Select Columns in Dataset’ module by double-clicking and entering comment text ‘Features f0…f783’, click arrow to show comment.

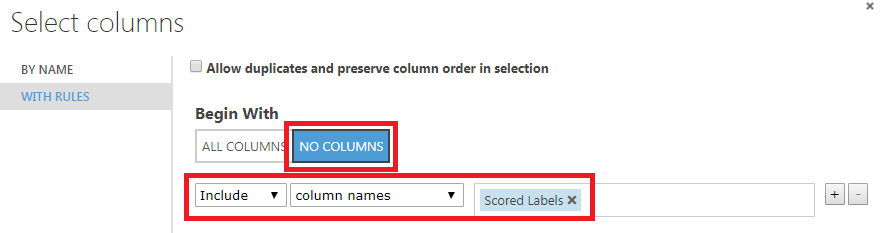
For other ‘Select Columns in Dataset’ enter comment text ‘Label (digit)’



1. Select module ‘Features f0…f783’ and in its Properties pane select ‘Launch column selector’
2. From ‘available columns’ select all the f0..f783 columns and move them across to the ‘SELECTED COLUMNS’ then tick



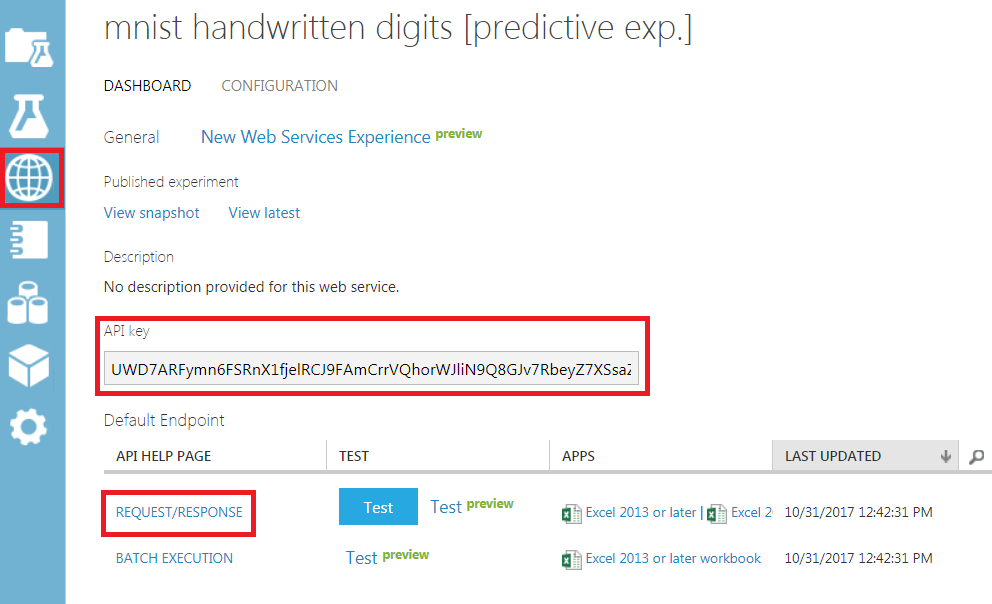
1. Select module ‘Label (digit)’ and in its Properties pane select ‘Launch column selector’
2. Select ‘Begins With NO COLUMNS’, ‘Include column names’ then ‘Scored Labels’, tick



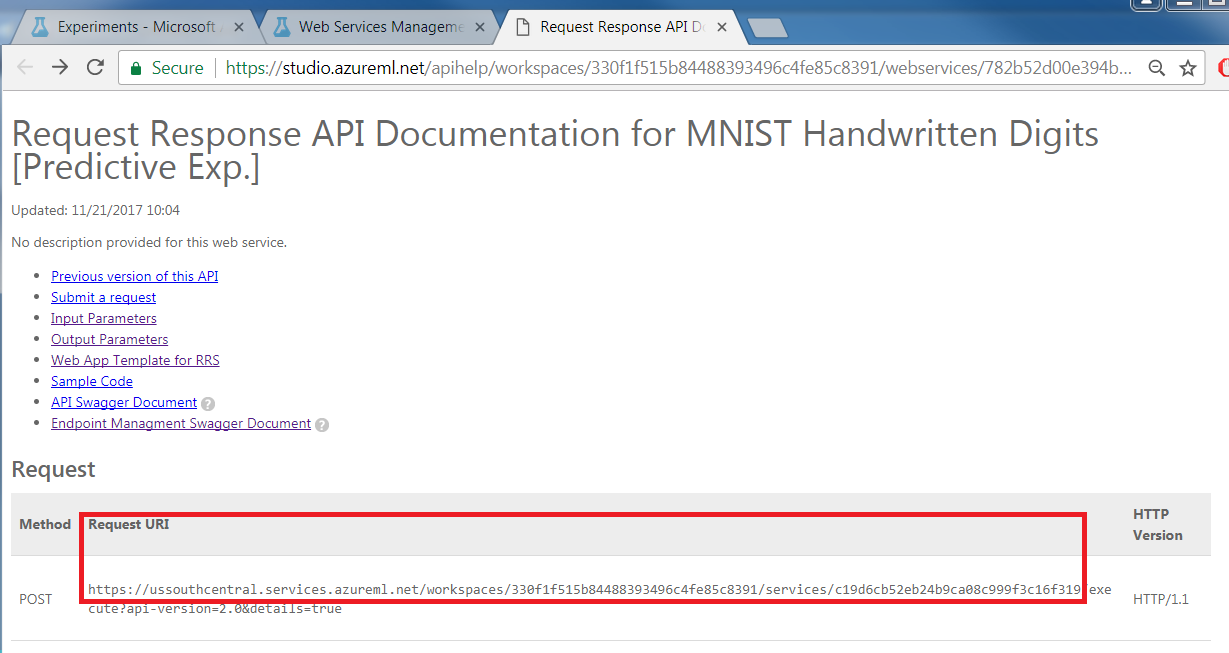
1. Select ‘Run’ to validate changes
2. Click ‘DEPLOY WEB SERVICE’



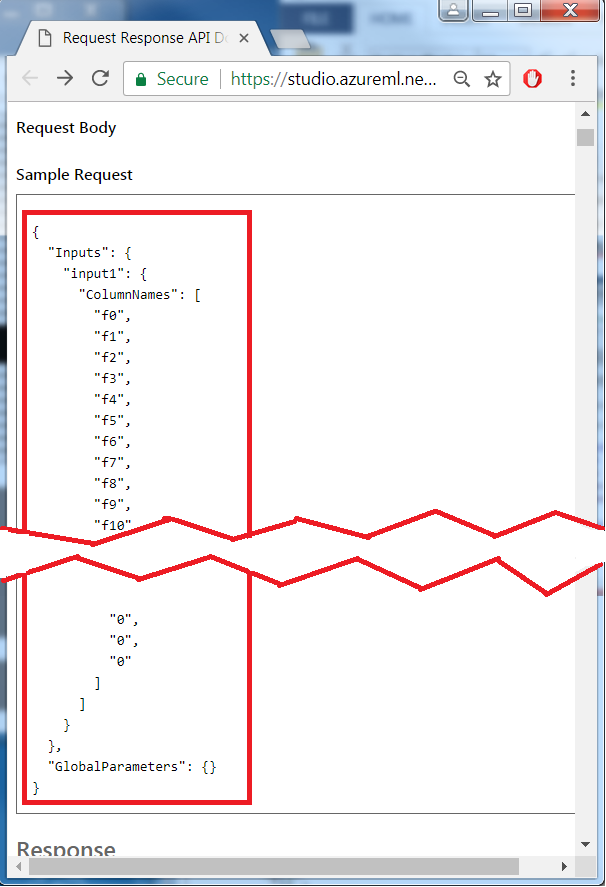
1. A new Web Service pane should now appear. Take a note of the API key (let’s call it ML\_Web\_Service\_API\_Key for later use)



1. Click on ‘REQUEST/RESPONSE’
2. A new browser tab ‘Request Response API Documentation for MNIST …’ opens. Take a note of the Request URI up to but not including ‘/execute?api-version=2.0&details=true’ (for later use ML\_Web\_Service\_Request\_URI)



1. Scroll down until you get to Sample Request. Take a note of the sample request for later user ML\_Web\_Service\_Sample\_Request (note this is very long over 1500 lines)



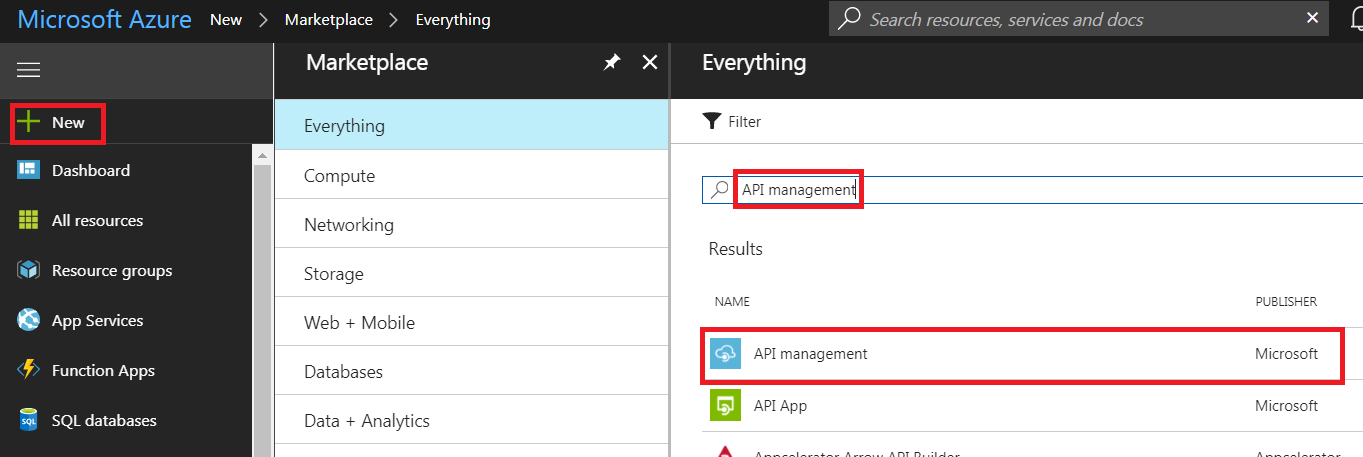
# Create a API under API management service

In the section you will create a new publicly accessible API inside Azure’s API management service which will call the MNIST ML Web Service

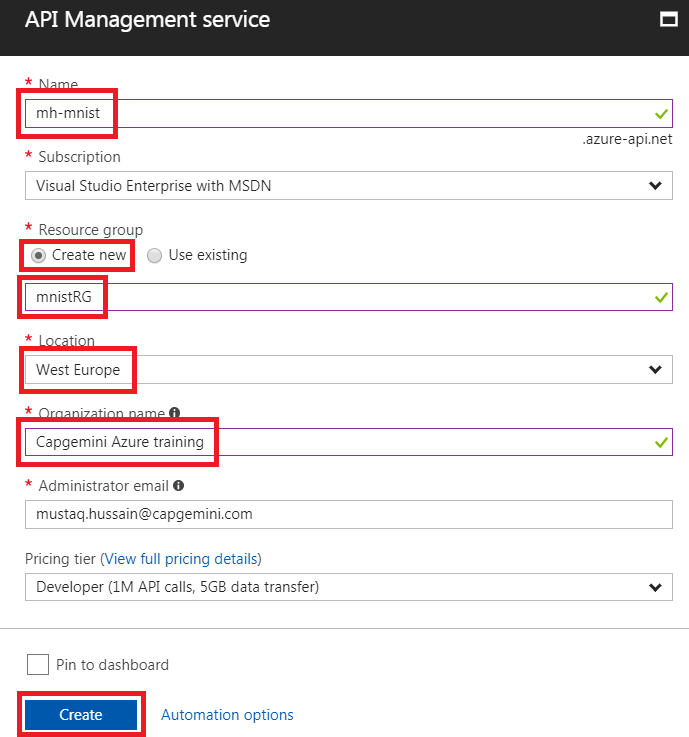
1. Login to your Azure portal

https://portal.azure.com

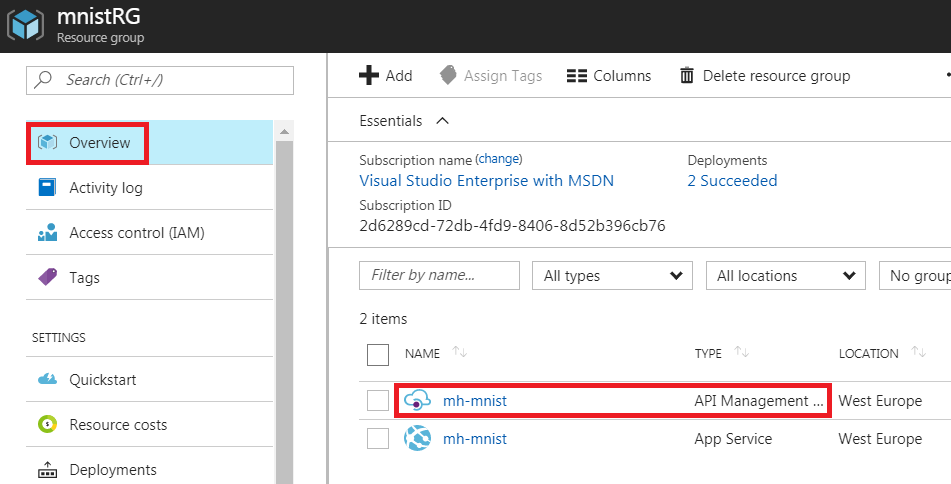
1. (NOTE: Skip this and next step if you have created API management) Click ‘+NEW’->‘API management’->‘API management’->’Create’



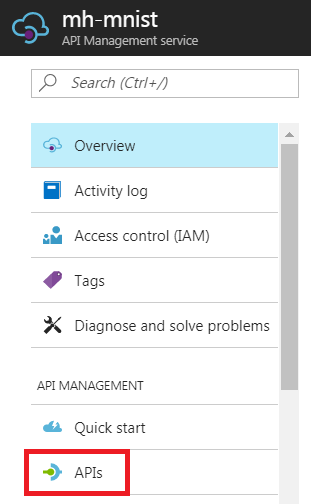
1. Then fill in the API Management service details as below (use your own initials/name) and press ‘Create’ – creation will take ~30 minutes.



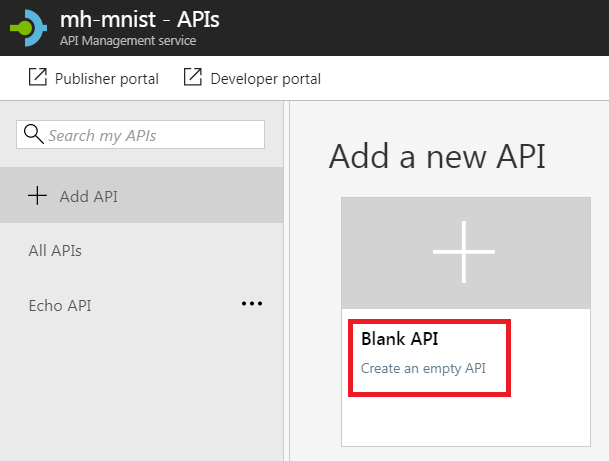
1. From Resource Group select newly created API Management service, ‘mnistRG’->mh-minist



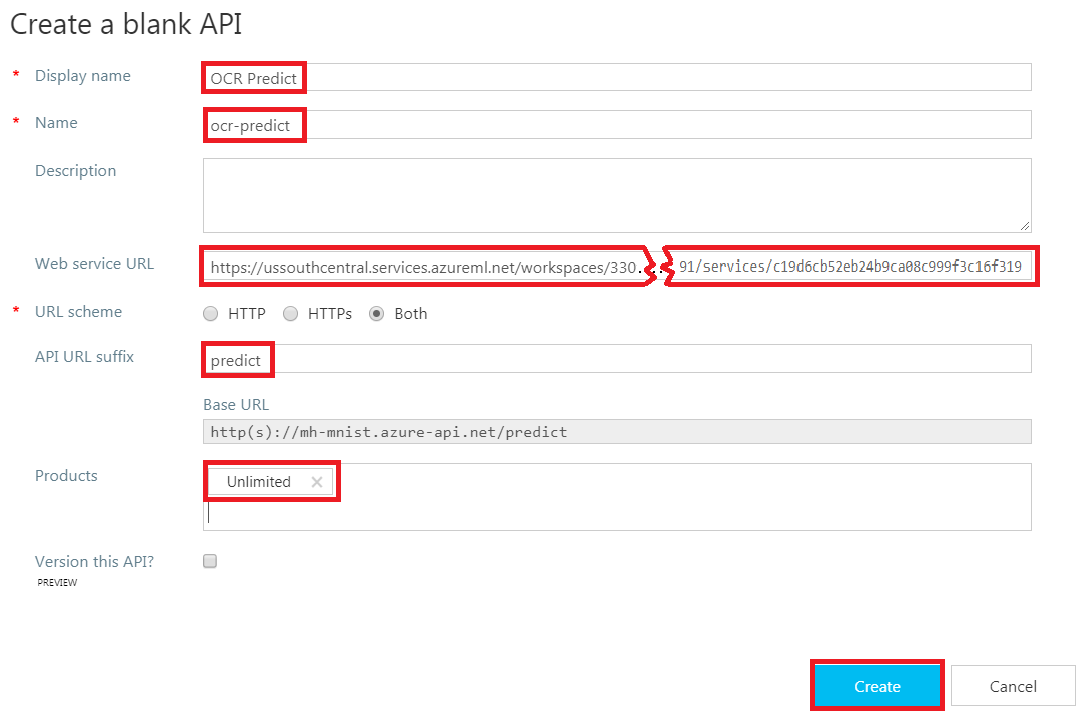
1. Select ‘APIs’ from API Management service



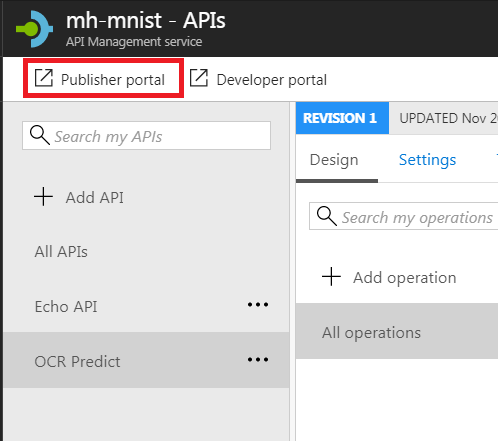
1. Select ‘Create an empty API’



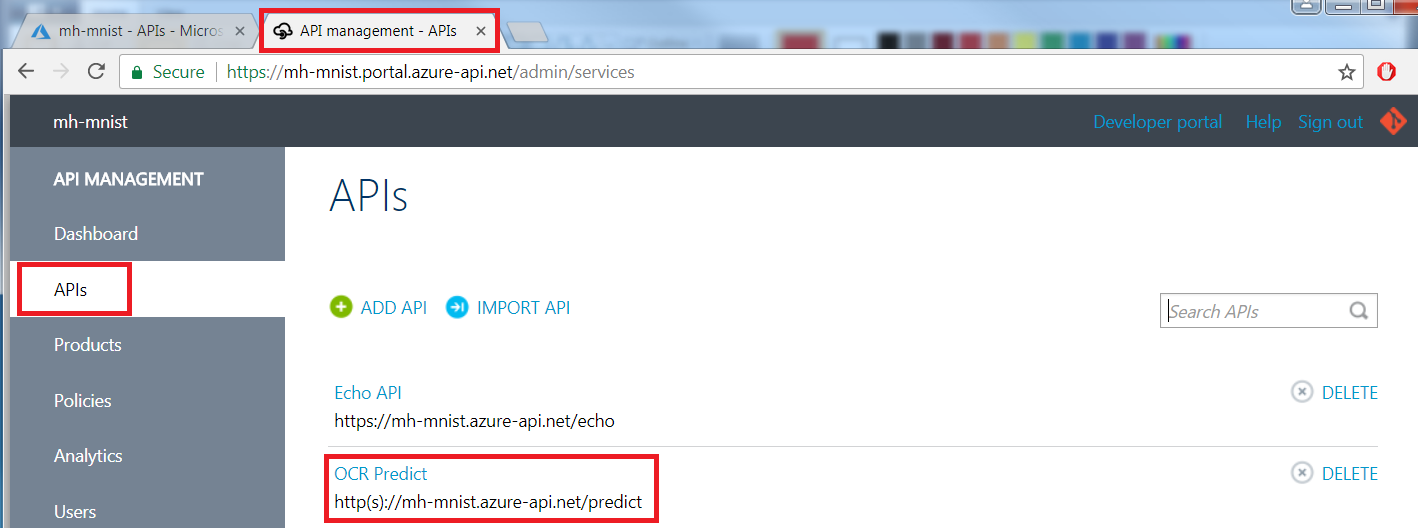
1. Fill in the API details as below (with Web server Request URI up to but not including ‘/execute?api-version=2.0&details=true’ from previous section ML\_Web\_Service\_Request\_URI) then ‘Create’



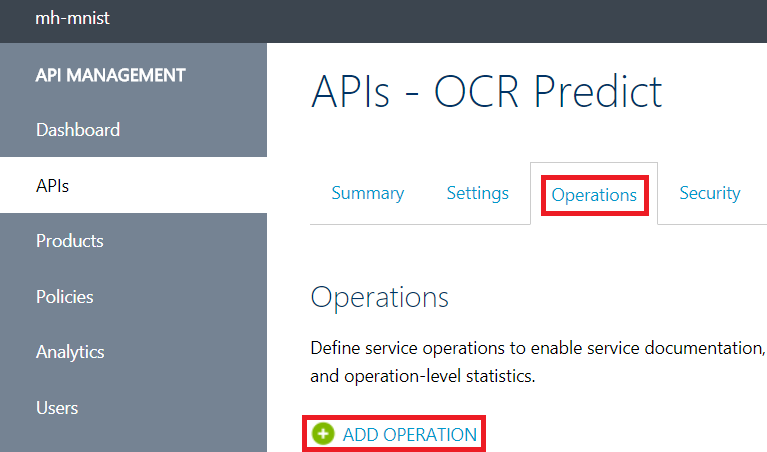
1. Back at API management service - APIs press ‘Publisher portal’



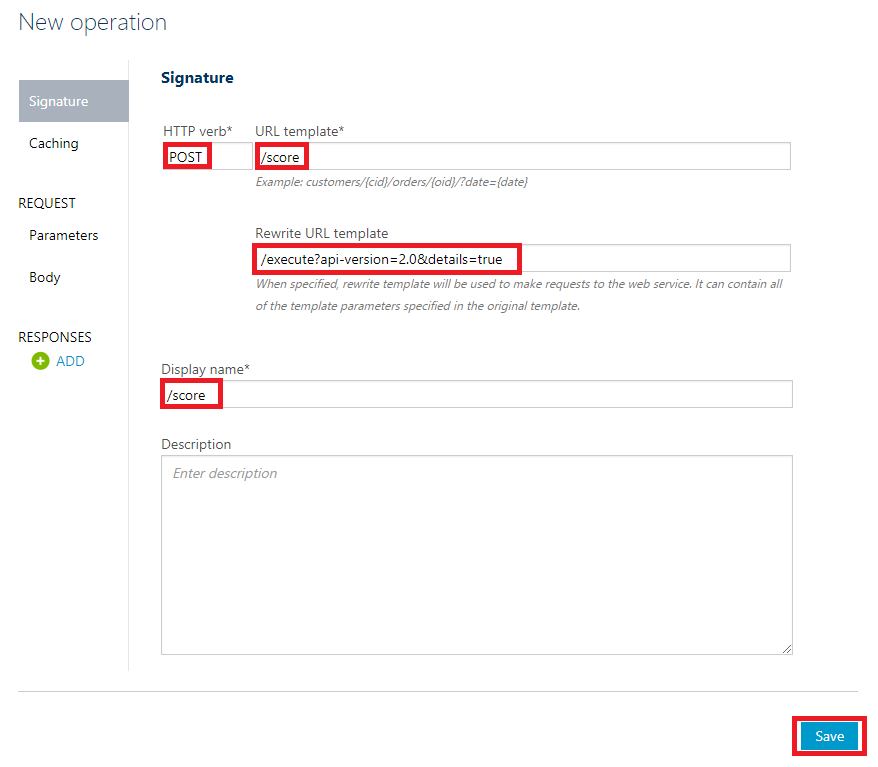
1. A new browser tab ‘API Management – APIs’ will appear. Select ‘APIs’->’OCR Predict’



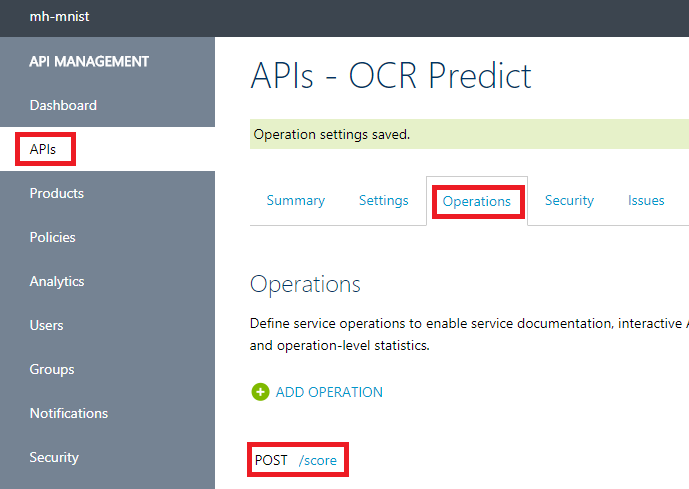
1. Select ‘Operations’->’ADD OPERATION’



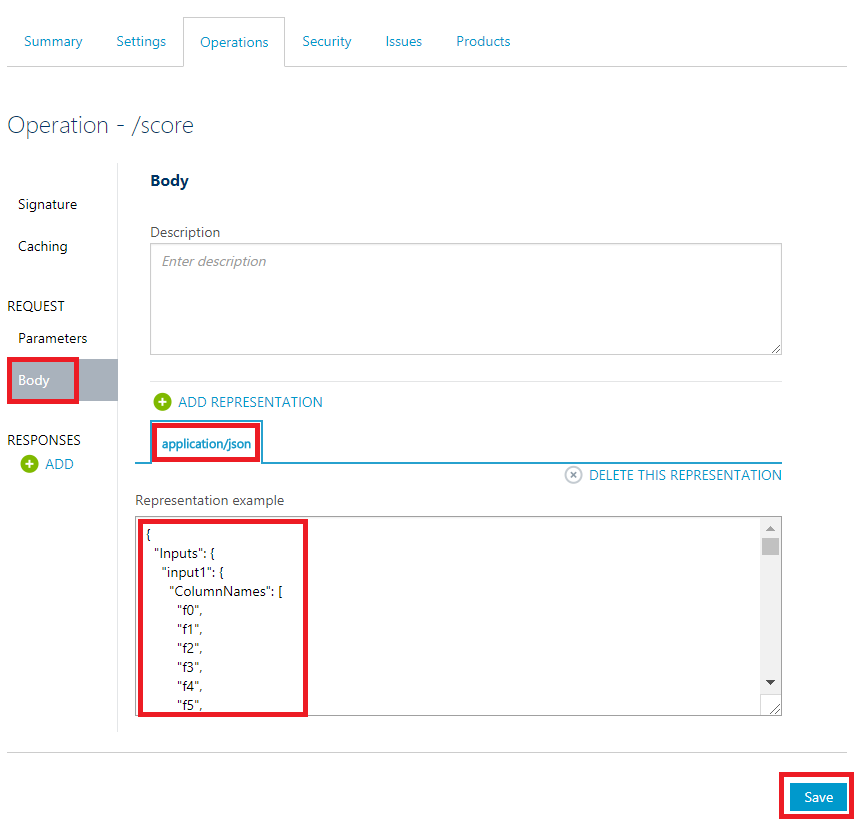
1. Enter following details



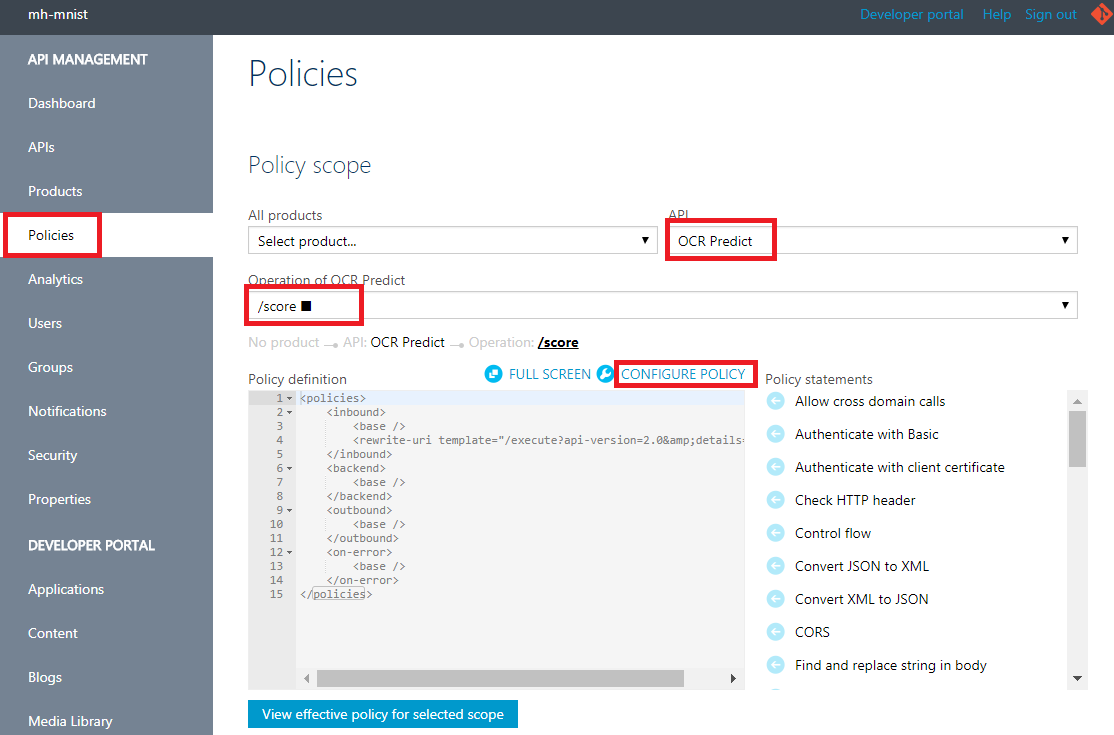
1. Select ‘POST /score’



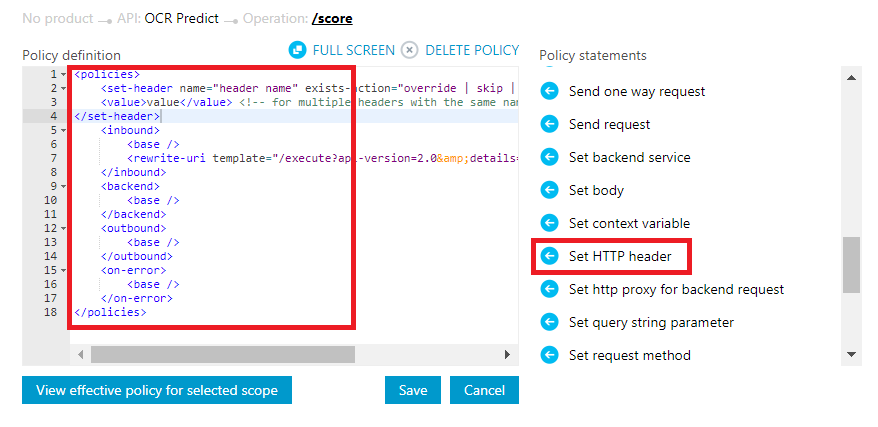
1. Select ‘Body’ and enter ‘Application/Json’ and paste the very long ML\_Web\_Service\_Sample\_Request from previous section and ‘Save’



1. Select ‘Polices’->’OCR Predict’->’/score’->’CONFIGURE POLICY’



1. Under ‘Policy statements’ scroll down and select ‘Set HTTP header’



1. Replace Policy definition with following code , replacing ML\_Web\_Service\_API\_Key with your own key

<policies>

<inbound>

<base />

<cors>

<allowed-origins>

<origin>\*</origin>

<!-- allow any -->

</allowed-origins>

<allowed-methods>

<!-- allow any -->

<method>\*</method>

</allowed-methods>

<allowed-headers>

<!-- allow any -->

<header>\*</header>

</allowed-headers>

</cors>

<set-header name="Authorization" exists-action="override">

<value>Bearer **ML\_Web\_Service\_API\_Key**</value>

</set-header>

<rewrite-uri template="/execute?api-version=2.0&amp;details=true" />

</inbound>

<backend>

<base />

</backend>

<outbound>

<base />

</outbound>

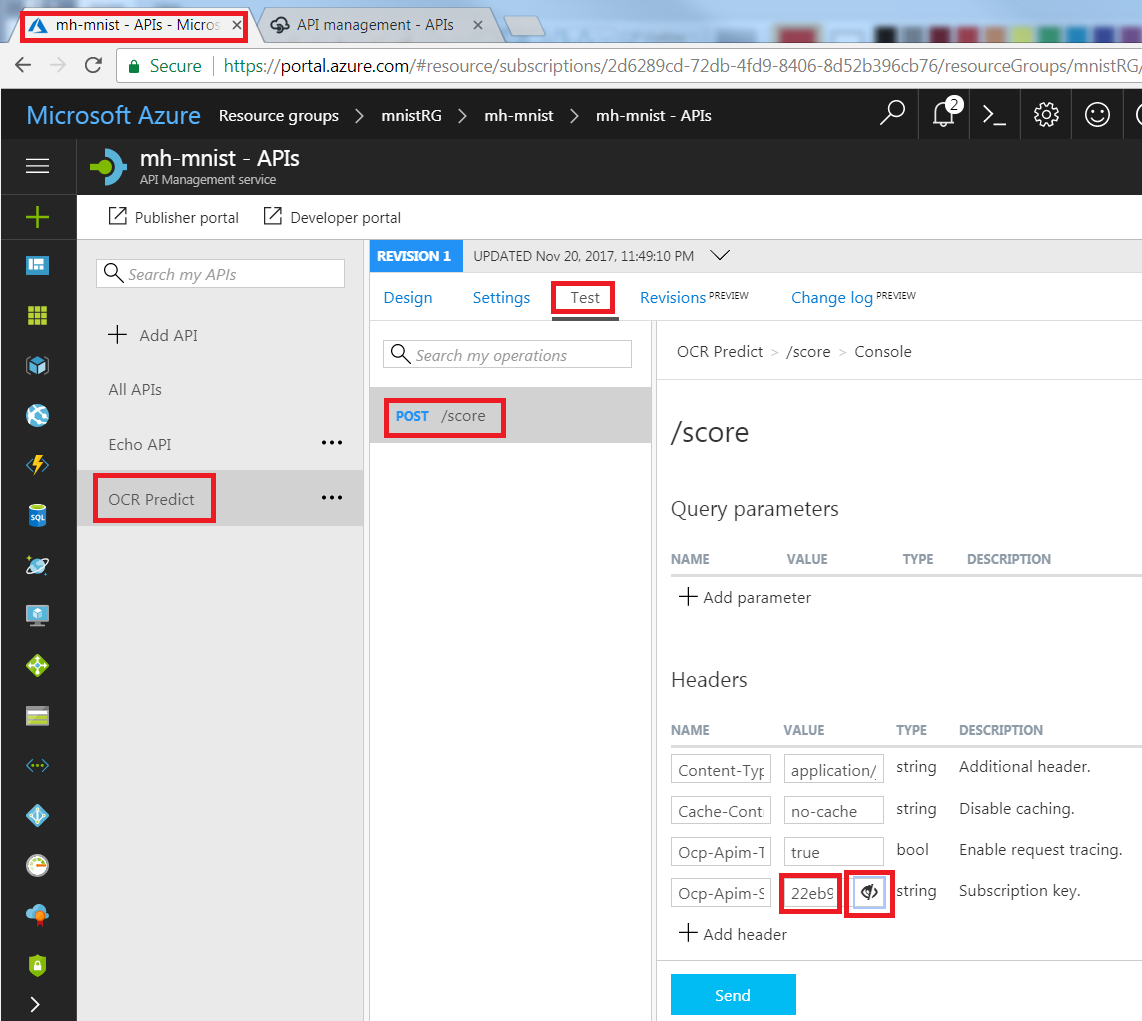
<on-error>

<base />

</on-error>

</policies>

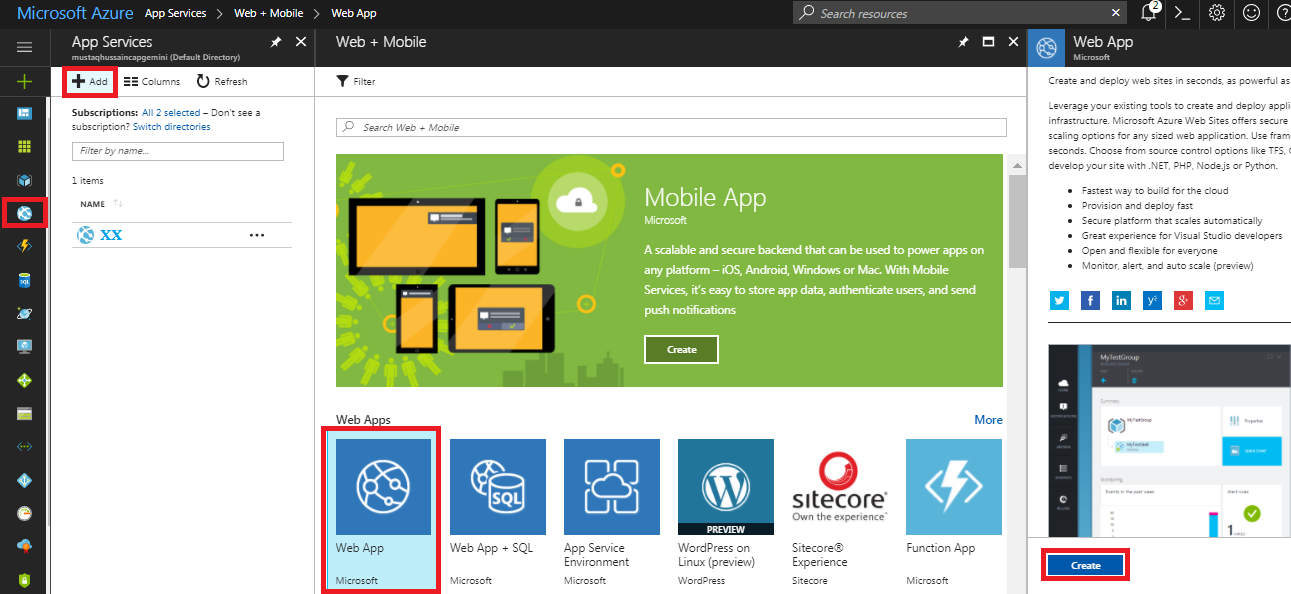
1. Go back to portal tab and API management service-API and select ‘OCR Predict’->’POST /score’->’Test’->’Show/Hide’ Subscription key and take a note of the OCP\_APIM\_SUBSCRIPTION\_KEY



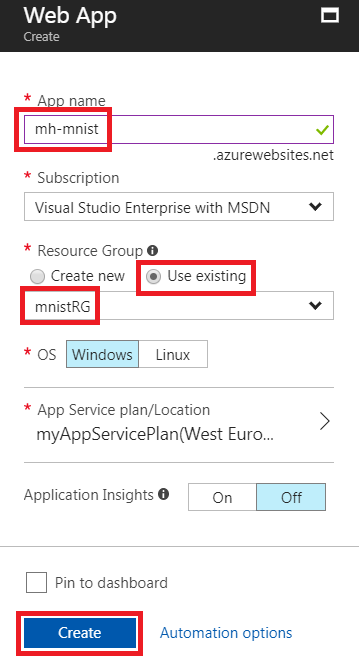
# Create New App Service Web App

In this section you will create a new Azure Web App as an App Service where the handwritten digit application will be deployed

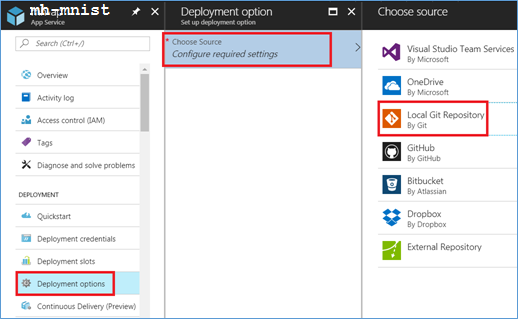
1. Select ‘App Services’->’Add’->’Web App’->’Create’



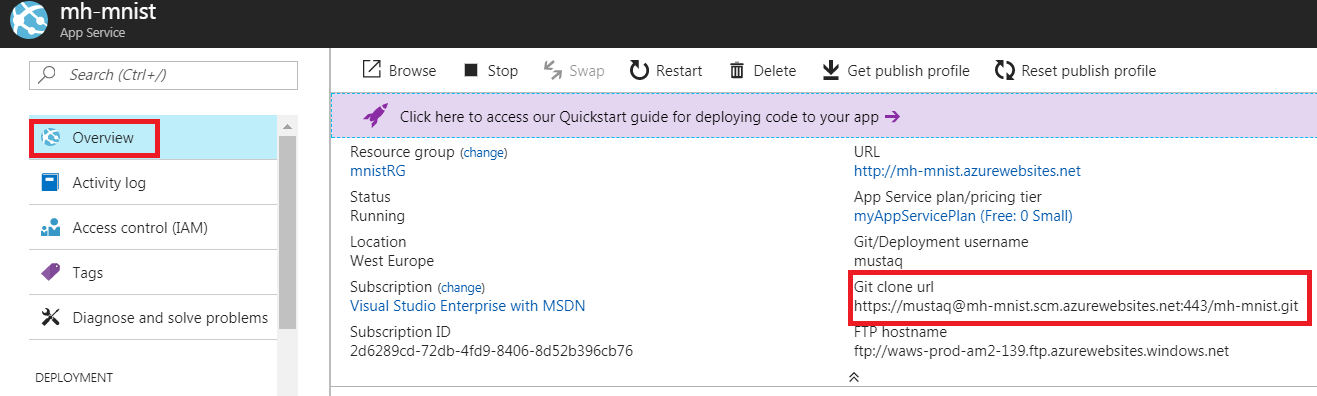
1. Fill in ‘Web App’ details followed by Create



1. Close all blades
2. Select ‘App Services’-> ‘mh-mnist’
3. Select ‘Deployment options’->’Configure required settings’->’Local Git Repository’ ->’OK’



1. Select ‘Overview’ and note the Git clone URL



(note in screen shot username ‘mustaq’ inherited from previous Library App Service sessions)

# Create Locally Handwritten Digit Web Application

In this section you will create a Browser based application, where you can draw a digit and submit it via a restful http call to the newly created mnist API function (which calls your mnist ML Web Service)

1. On your laptop create an empty folder at d:\projects\vs\mnist
2. Create new file “index.html” in the mnist directory
3. Copy and save following into “index.html”

<html>

<head>

<title>Optical Character Recognition with Microsoft Azure Machine Learning</title>

<style>

#imgView {

border: 1px solid #FFFFFF;

}

#btnClear {

background: gray;

width: 280px;

color: white;

font-size: 2em;

}

#btnSend {

background: gray;

width: 280px;

color: white;

font-size: 2em;

}

</style>

</head>

<body>

<canvas id="imgView" width="280" height="280">

Unfortunately, your browser does not supported.

</canvas>

<p><button id="btnClear">Clear</button></p>

<p><button id="btnSend">Send</button></p>

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

<script src="./odr.js"></script>

</body>

</html>

1. Create another new file “odr.js” in the mnist directory
2. Copy and save following into “odr.js”

window.addEventListener('load', function () {

var canvas, context;

var iw = 28; // image width

var ih = 28; // image height

var s = 10; // scale (in HTML the canvas size is 280px, mapping canvas size to MINST sample size 28x28)

var dimg; // digit pixels in 2D array

function init() {

canvas = document.getElementById('imgView');

context = canvas.getContext('2d');

drawtool = new canvasPencil();

canvas.addEventListener('mousedown', drawtool.mousedown, false);

canvas.addEventListener('mousemove', drawtool.mousemove, false);

canvas.addEventListener('mouseup', drawtool.mouseup, false);

canvas.addEventListener('mouseleave', drawtool.mouseup, false);

var btnClear = document.getElementById("btnClear");

btnClear.addEventListener("click", clearCanvas, false);

var btnSend = document.getElementById("btnSend");

btnSend.addEventListener("click", callWebService, false);

drawCanvasGrid();

dimg = new Array(ih);

for (var i = 0; i < ih; i++) {

dimg[i] = new Array(iw);

}

clearCanvas();

}

function canvasPencil() {

var isMouseDown = false;

var mouseX = 0;

var mouseY = 0;

this.mousedown = function (evt) {

isMouseDown = true;

mouseX = evt.offsetX;

mouseY = evt.offsetY;

context.beginPath();

context.moveTo(mouseX, mouseY);

};

this.mousemove = function (evt) {

if (isMouseDown) {

mouseX = evt.offsetX;

mouseY = evt.offsetY;

dx = Math.floor(mouseX / 10);

dy = Math.floor(mouseY / 10);

drawCanvasCell(dx, dy); // draw pixel on the canvas

dimg[dx][dy] = 1; // set the same pixel on 2D array

// not a thin line, but a bold line, like the original digit drawings...

if (dx < 27 && dy < 27 && dx > 0 && dy > 0){

drawCanvasCell(dx, dy + 1);

drawCanvasCell(dx, dy - 1);

drawCanvasCell(dx + 1, dy + 1);

drawCanvasCell(dx + 1, dy);

drawCanvasCell(dx + 1, dy - 1);

drawCanvasCell(dx - 1, dy + 1);

drawCanvasCell(dx - 1, dy);

drawCanvasCell(dx - 1, dy - 1);

// not a thin line, but a bold line, like the original digit drawings...

dimg[dx][dy + 1] = 1;

dimg[dx][dy - 1] = 1;

dimg[dx + 1][dy + 1] = 1;

dimg[dx + 1][dy] = 1;

dimg[dx + 1][dy - 1] = 1;

dimg[dx - 1][dy + 1] = 1;

dimg[dx - 1][dy] = 1;

dimg[dx - 1][dy - 1] = 1;

}

}

};

this.mouseup = function (evt) {

isMouseDown = false;

};

}

function clearCanvas() {

canvas.width = canvas.width;

drawCanvasGrid();

for (var i = 0; i < ih; i++) {

for (var j = 0; j < iw; j++) {

dimg[i][j] = 0;

}

}

}

function drawCanvasCell(x, y) {

context.fillRect(x \* s, y \* s, s, s);

}

function drawCanvasGrid() {

for (var x = 0; x <= iw; x += 1) {

context.moveTo(x \* s, 0);

context.lineTo(x \* s, ih \* s);

}

for (var y = 0; y <= ih; y += 1) {

context.moveTo(0, y \* s);

context.lineTo(iw \* s, y \* s);

}

context.stroke();

}

function callWebService() {

var dimgarray = ""

for (var i = 0; i < ih; i++) {

for (var j = 0; j < iw; j++) {

dimgarray += dimg[j][i] + ",";

}

}

dimgarray = dimgarray.slice(0, dimgarray.length - 1);

var arg = {

"Inputs": {

"input1": {

"ColumnNames": ["f0", "f1", "f2", "f3", "f4", "f5", "f6", "f7", "f8", "f9", "f10", "f11", "f12", "f13", "f14", "f15", "f16", "f17", "f18", "f19", "f20", "f21", "f22", "f23", "f24", "f25", "f26", "f27", "f28", "f29", "f30", "f31", "f32", "f33", "f34", "f35", "f36", "f37", "f38", "f39", "f40", "f41", "f42", "f43", "f44", "f45", "f46", "f47", "f48", "f49", "f50", "f51", "f52", "f53", "f54", "f55", "f56", "f57", "f58", "f59", "f60", "f61", "f62", "f63", "f64", "f65", "f66", "f67", "f68", "f69", "f70", "f71", "f72", "f73", "f74", "f75", "f76", "f77", "f78", "f79", "f80", "f81", "f82", "f83", "f84", "f85", "f86", "f87", "f88", "f89", "f90", "f91", "f92", "f93", "f94", "f95", "f96", "f97", "f98", "f99", "f100", "f101", "f102", "f103", "f104", "f105", "f106", "f107", "f108", "f109", "f110", "f111", "f112", "f113", "f114", "f115", "f116", "f117", "f118", "f119", "f120", "f121", "f122", "f123", "f124", "f125", "f126", "f127", "f128", "f129", "f130", "f131", "f132", "f133", "f134", "f135", "f136", "f137", "f138", "f139", "f140", "f141", "f142", "f143", "f144", "f145", "f146", "f147", "f148", "f149", "f150", "f151", "f152", "f153", "f154", "f155", "f156", "f157", "f158", "f159", "f160", "f161", "f162", "f163", "f164", "f165", "f166", "f167", "f168", "f169", "f170", "f171", "f172", "f173", "f174", "f175", "f176", "f177", "f178", "f179", "f180", "f181", "f182", "f183", "f184", "f185", "f186", "f187", "f188", "f189", "f190", "f191", "f192", "f193", "f194", "f195", "f196", "f197", "f198", "f199", "f200", "f201", "f202", "f203", "f204", "f205", "f206", "f207", "f208", "f209", "f210", "f211", "f212", "f213", "f214", "f215", "f216", "f217", "f218", "f219", "f220", "f221", "f222", "f223", "f224", "f225", "f226", "f227", "f228", "f229", "f230", "f231", "f232", "f233", "f234", "f235", "f236", "f237", "f238", "f239", "f240", "f241", "f242", "f243", "f244", "f245", "f246", "f247", "f248", "f249", "f250", "f251", "f252", "f253", "f254", "f255", "f256", "f257", "f258", "f259", "f260", "f261", "f262", "f263", "f264", "f265", "f266", "f267", "f268", "f269", "f270", "f271", "f272", "f273", "f274", "f275", "f276", "f277", "f278", "f279", "f280", "f281", "f282", "f283", "f284", "f285", "f286", "f287", "f288", "f289", "f290", "f291", "f292", "f293", "f294", "f295", "f296", "f297", "f298", "f299", "f300", "f301", "f302", "f303", "f304", "f305", "f306", "f307", "f308", "f309", "f310", "f311", "f312", "f313", "f314", "f315", "f316", "f317", "f318", "f319", "f320", "f321", "f322", "f323", "f324", "f325", "f326", "f327", "f328", "f329", "f330", "f331", "f332", "f333", "f334", "f335", "f336", "f337", "f338", "f339", "f340", "f341", "f342", "f343", "f344", "f345", "f346", "f347", "f348", "f349", "f350", "f351", "f352", "f353", "f354", "f355", "f356", "f357", "f358", "f359", "f360", "f361", "f362", "f363", "f364", "f365", "f366", "f367", "f368", "f369", "f370", "f371", "f372", "f373", "f374", "f375", "f376", "f377", "f378", "f379", "f380", "f381", "f382", "f383", "f384", "f385", "f386", "f387", "f388", "f389", "f390", "f391", "f392", "f393", "f394", "f395", "f396", "f397", "f398", "f399", "f400", "f401", "f402", "f403", "f404", "f405", "f406", "f407", "f408", "f409", "f410", "f411", "f412", "f413", "f414", "f415", "f416", "f417", "f418", "f419", "f420", "f421", "f422", "f423", "f424", "f425", "f426", "f427", "f428", "f429", "f430", "f431", "f432", "f433", "f434", "f435", "f436", "f437", "f438", "f439", "f440", "f441", "f442", "f443", "f444", "f445", "f446", "f447", "f448", "f449", "f450", "f451", "f452", "f453", "f454", "f455", "f456", "f457", "f458", "f459", "f460", "f461", "f462", "f463", "f464", "f465", "f466", "f467", "f468", "f469", "f470", "f471", "f472", "f473", "f474", "f475", "f476", "f477", "f478", "f479", "f480", "f481", "f482", "f483", "f484", "f485", "f486", "f487", "f488", "f489", "f490", "f491", "f492", "f493", "f494", "f495", "f496", "f497", "f498", "f499", "f500", "f501", "f502", "f503", "f504", "f505", "f506", "f507", "f508", "f509", "f510", "f511", "f512", "f513", "f514", "f515", "f516", "f517", "f518", "f519", "f520", "f521", "f522", "f523", "f524", "f525", "f526", "f527", "f528", "f529", "f530", "f531", "f532", "f533", "f534", "f535", "f536", "f537", "f538", "f539", "f540", "f541", "f542", "f543", "f544", "f545", "f546", "f547", "f548", "f549", "f550", "f551", "f552", "f553", "f554", "f555", "f556", "f557", "f558", "f559", "f560", "f561", "f562", "f563", "f564", "f565", "f566", "f567", "f568", "f569", "f570", "f571", "f572", "f573", "f574", "f575", "f576", "f577", "f578", "f579", "f580", "f581", "f582", "f583", "f584", "f585", "f586", "f587", "f588", "f589", "f590", "f591", "f592", "f593", "f594", "f595", "f596", "f597", "f598", "f599", "f600", "f601", "f602", "f603", "f604", "f605", "f606", "f607", "f608", "f609", "f610", "f611", "f612", "f613", "f614", "f615", "f616", "f617", "f618", "f619", "f620", "f621", "f622", "f623", "f624", "f625", "f626", "f627", "f628", "f629", "f630", "f631", "f632", "f633", "f634", "f635", "f636", "f637", "f638", "f639", "f640", "f641", "f642", "f643", "f644", "f645", "f646", "f647", "f648", "f649", "f650", "f651", "f652", "f653", "f654", "f655", "f656", "f657", "f658", "f659", "f660", "f661", "f662", "f663", "f664", "f665", "f666", "f667", "f668", "f669", "f670", "f671", "f672", "f673", "f674", "f675", "f676", "f677", "f678", "f679", "f680", "f681", "f682", "f683", "f684", "f685", "f686", "f687", "f688", "f689", "f690", "f691", "f692", "f693", "f694", "f695", "f696", "f697", "f698", "f699", "f700", "f701", "f702", "f703", "f704", "f705", "f706", "f707", "f708", "f709", "f710", "f711", "f712", "f713", "f714", "f715", "f716", "f717", "f718", "f719", "f720", "f721", "f722", "f723", "f724", "f725", "f726", "f727", "f728", "f729", "f730", "f731", "f732", "f733", "f734", "f735", "f736", "f737", "f738", "f739", "f740", "f741", "f742", "f743", "f744", "f745", "f746", "f747", "f748", "f749", "f750", "f751", "f752", "f753", "f754", "f755", "f756", "f757", "f758", "f759", "f760", "f761", "f762", "f763", "f764", "f765", "f766", "f767", "f768", "f769", "f770", "f771", "f772", "f773", "f774", "f775", "f776", "f777", "f778", "f779", "f780", "f781", "f782", "f783"],

"Values": [ ]

}

},

"GlobalParameters": {}

}

p = JSON.parse("[" + dimgarray + "]");

arg.Inputs.input1.Values.push(p);

jQuery.ajax({

url: "https://API\_Management\_Service\_Name\_TODO.azure-api.net/predict/score",

beforeSend: function (xhrObj) {

xhrObj.setRequestHeader("Content-Type", "application/json");

xhrObj.setRequestHeader("Ocp-Apim-Subscription-Key", "OCP\_APIM\_SUBSCRIPTION\_KEY");

},

type: "POST",

data: JSON.stringify(arg)

})

.done(function (data) {

res = data.Results.output1.value.Values

$.each(res, function (index, element) {

alert("Digit drawn: " + element)

});

})

.fail(function () {

alert("Error NOT GOING TO PREDICT");

});

}

init();

}, false);

1. Near the bottom of odr.js find text API\_Management\_Service\_Name\_TODO and replace text with the name of your previously created API (e.g. mh-mnist)
2. A few lines below that find OCP\_APIM\_SUBSCRIPTION\_KEY and replace with yours (from previous section)

# Deploy Application in Azure App Service

1. Open a command prompt window at d:\projects\vs\mnist
2. Before cloning this application to your App Service need to create a new local git repository:

git init

git add .

git commit -m "Check in: Initial Commit"

1. Create a remote link to your App Service Git clone URL and push the local code up to it (enter password when prompted for user)

git remote add mnist https://mustaq@mh-mnist.scm.azurewebsites.net:443/mh-mnist.git

git push mnist master

1. Finally test in a new browser with the url of your app

