Visual Studio LIVE! | San Diego

EXPERT SOLUTIONS FOR .NET DEVELOPERS

Using The Microsoft Cognitive Custom Vision Service

Michael Washington

Programmer

AiHelpWebsite.com

Level: Introductory/Intermediate

Code Again for the First Time!

Visual Studio 25 YEARS OF CODING INNOVATION

About Michael Washington

- Microsoft Reconnect MVP
- Microsoft Certified Professional

Microsoft Most Valuable Professional Reconnect

Microsoft CERTIFIED Professional

Visual Studio 25 YEARS OF CODING INNOVATION

Agenda

- Background
  - What is Custom Vision?
  - How to get started with Custom Vision (and how much it costs)
- Create A Custom Vision Application
  - Create a Custom Vision project that classifies images and allows you to upload images to be classified.
- .Net Core Angular Application (Part One)
  - Creating an application that allows you to upload an image and have it classified.
- .Net Core Angular Application (Part Two)
  - Upload new training images, tag them, and re-train the model.

What is Custom Vision?

- The Custom Vision Service is a Microsoft Cognitive Service that allows you build a custom image classifier
- The Custom Vision Service provides a web interface and a REST API that allows you to upload your images and train the classifier

Pricing details

The below pricing reflects a preview discount.

TIER	FEATURES	PRICE
Free	2 projects 5,000 training images per project 10,000 predictions	$0
Standard	100 projects Upload, training and prediction transactions	$1 per 1,000 transactions
	Image Storage (up to 6 MB each)	$0.35 per 1,000 images per month

How To Get Started With Custom Vision?

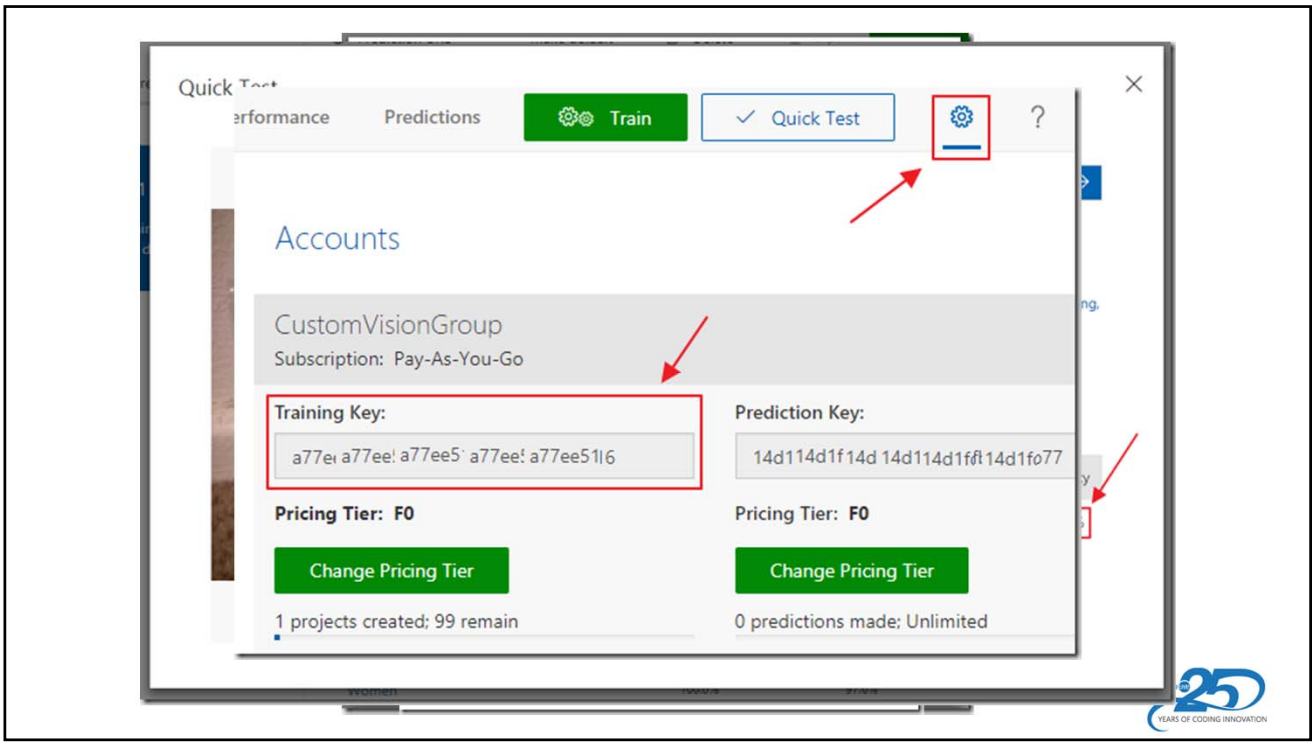
- A series of images to train your classifier (a minimum of 30 images per tag)

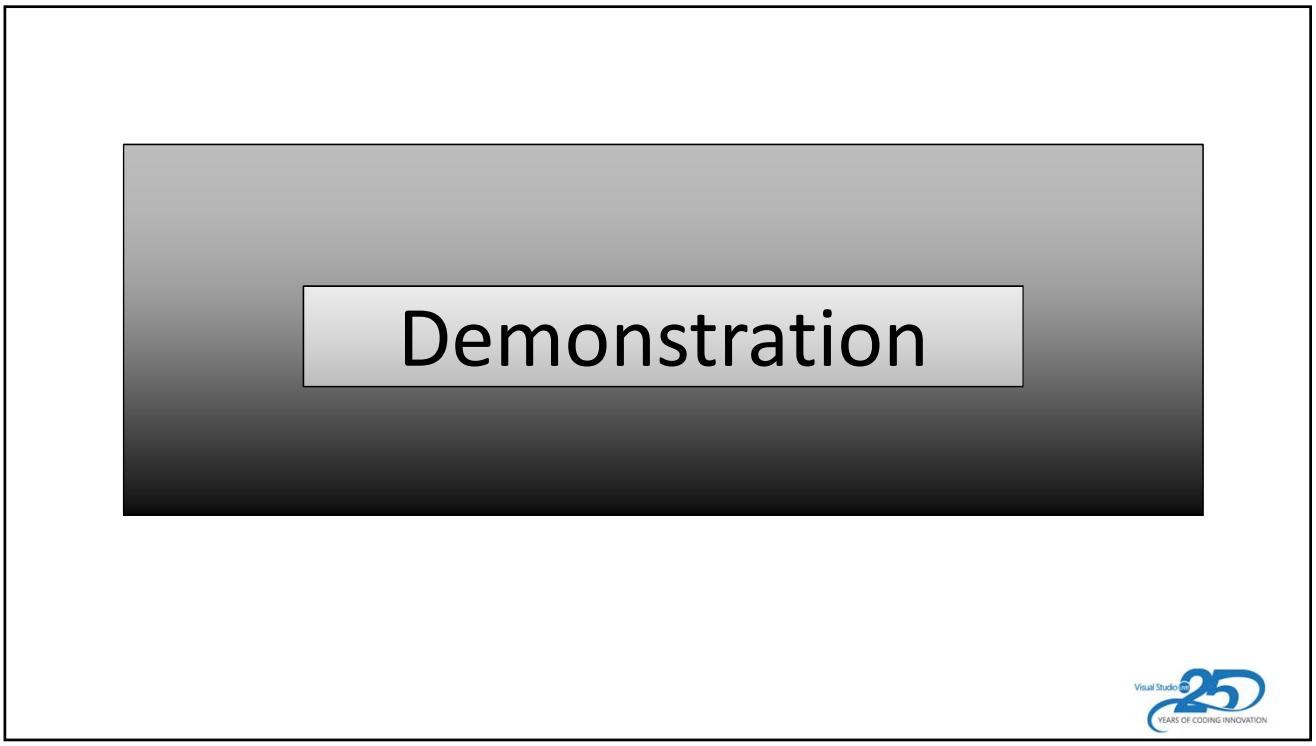
- A few images to test your classifier (after the classifier is trained)

- Optional: An Azure subscription (If you don't have an Azure subscription, you can create a free account before you begin)

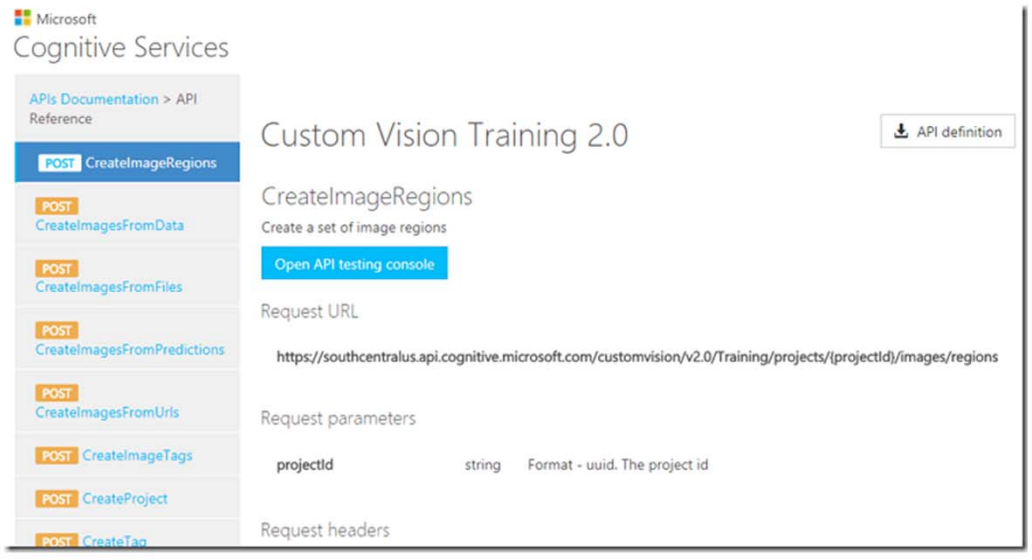
https://customvision.ai/

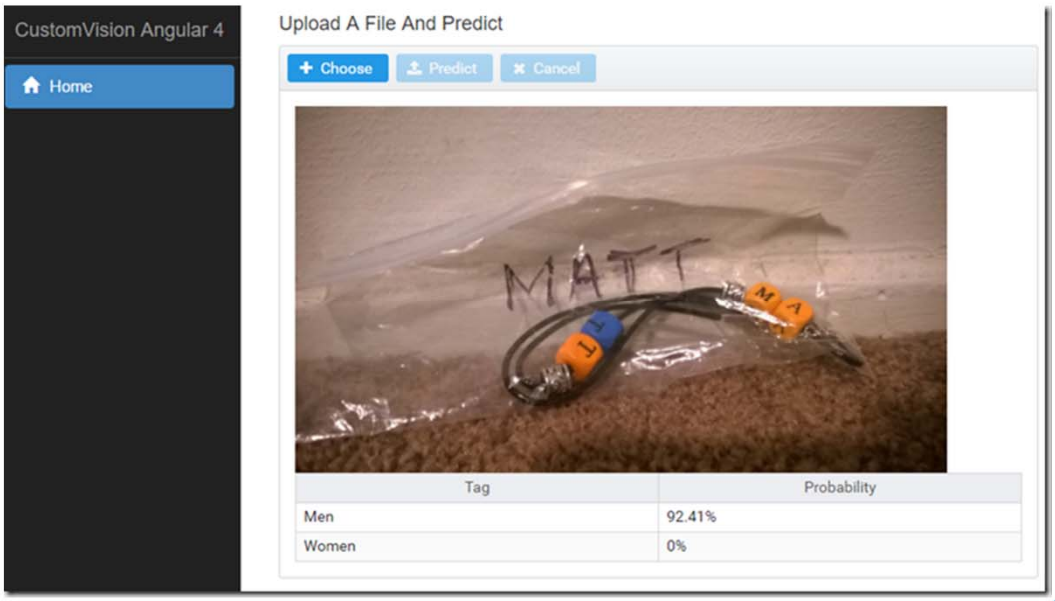
Create A Custom Vision Application


Performance
Predictions
Train
Quick Test
Accounts
CustomVisionGroup
Subscription: Pay-As-You-Go
Training Key:
Prediction Key:
Pricing Tier: F0
Pricing Tier: F0
Change Pricing Tier
Change Pricing Tier
1 projects created; 99 remain
0 predictions made; Unlimited


Demonstration

.Net Core Angular Application
(Part One)

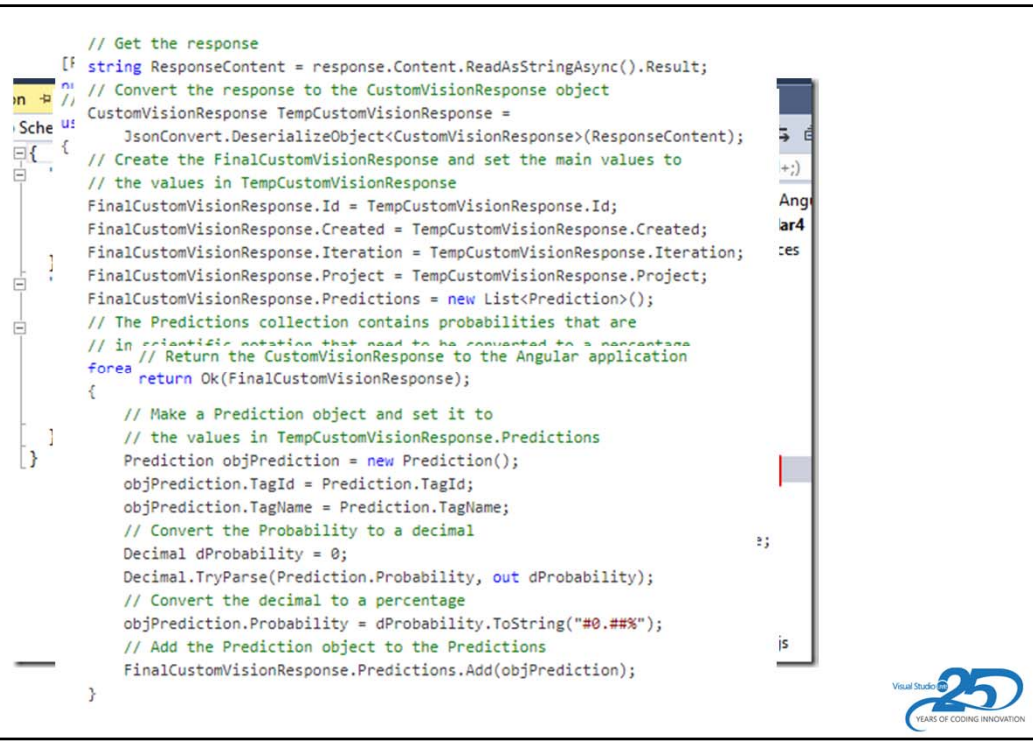




.Net Core Angular Application

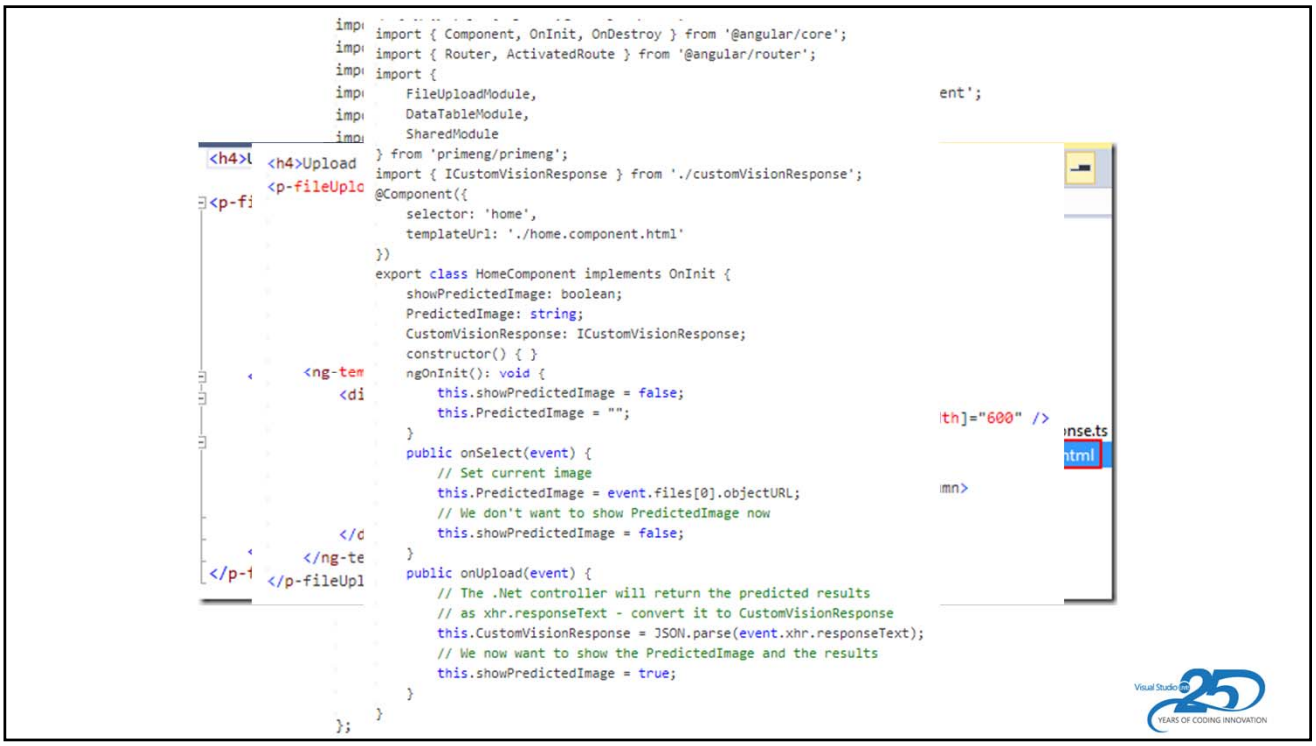
(Create The Application)

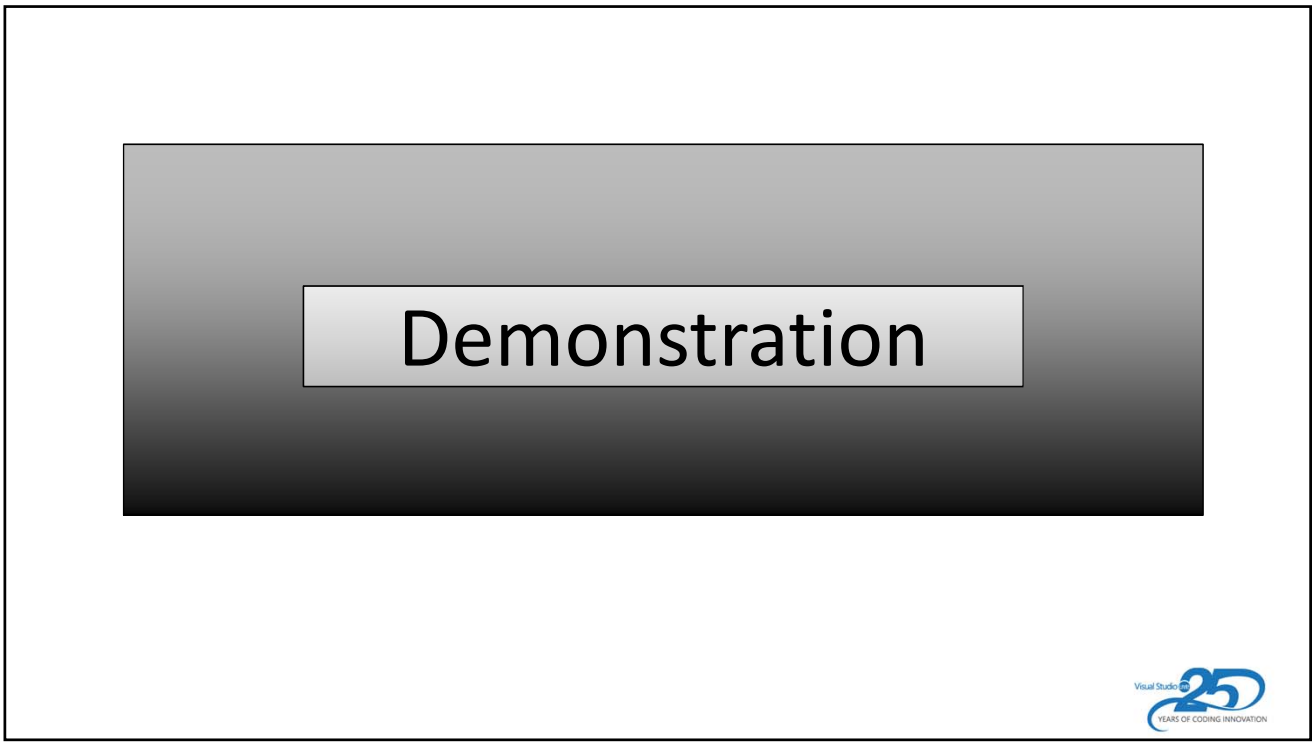
Server Side Code

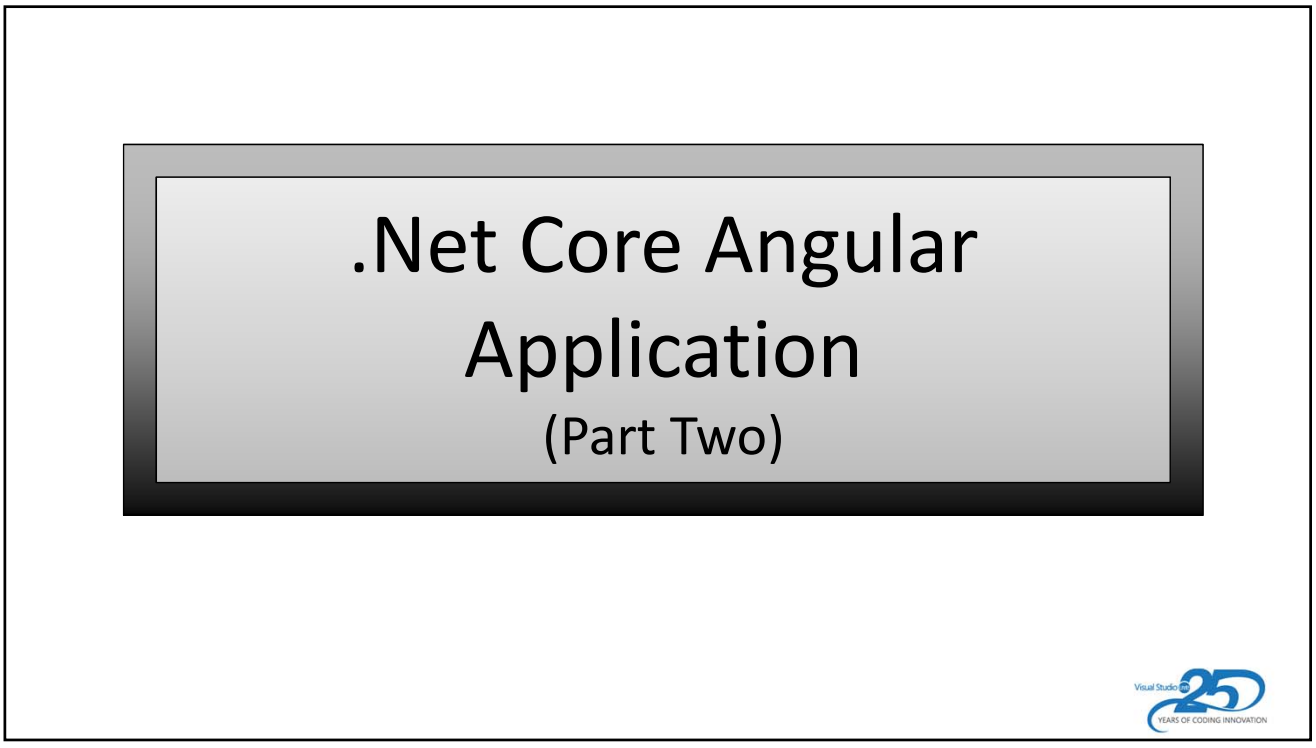


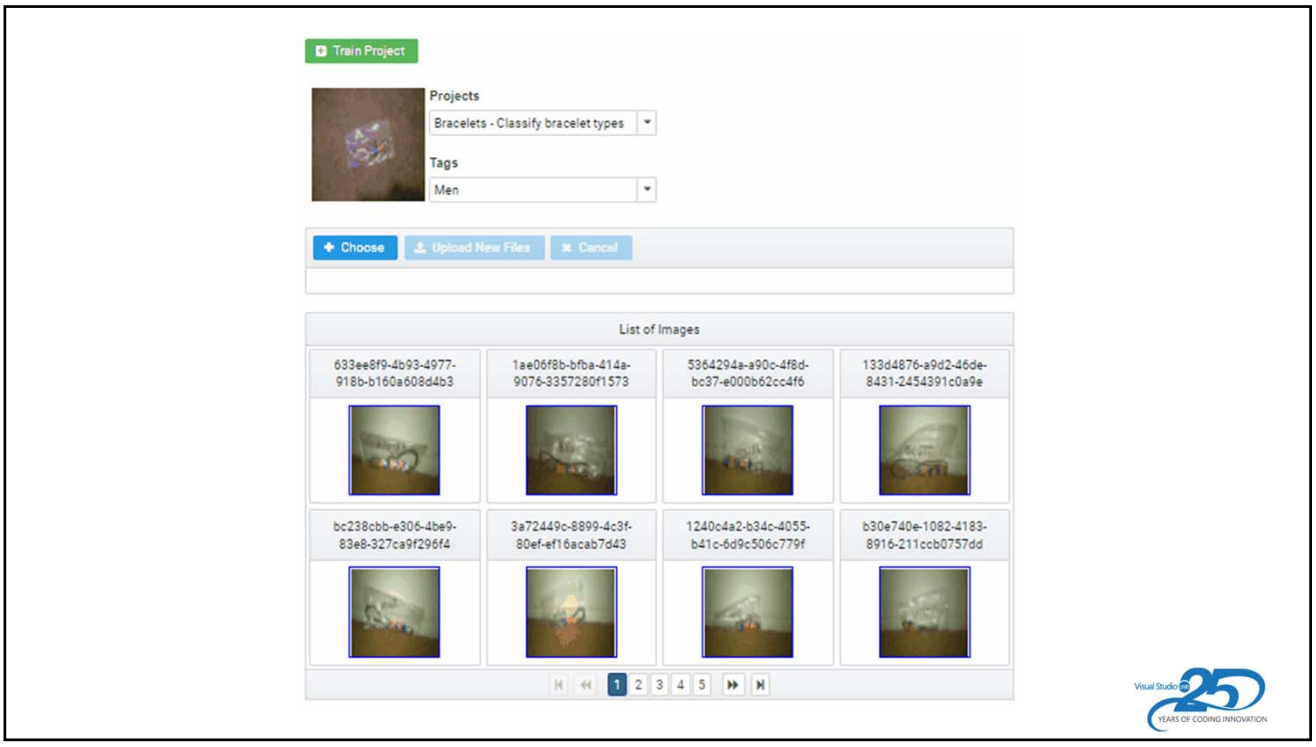
```
// Get the response
string ResponseContent = response.Content.ReadAsStringAsync().Result;
// Convert the response to the CustomVisionResponse object
CustomVisionResponse TempCustomVisionResponse =
    JsonConvert.DeserializeObject<CustomVisionResponse>(ResponseContent);
// Create the FinalCustomVisionResponse and set the main values to
// the values in TempCustomVisionResponse
FinalCustomVisionResponse.Id = TempCustomVisionResponse.Id;
FinalCustomVisionResponse.Created = TempCustomVisionResponse.Created;
FinalCustomVisionResponse.Iteration = TempCustomVisionResponse.Iteration;
FinalCustomVisionResponse.Project = TempCustomVisionResponse.Project;
FinalCustomVisionResponse.Predictions = new List<Prediction>();
// The Predictions collection contains probabilities that are
// in scientific notation that need to be converted to a percentage
// Return the CustomVisionResponse to the Angular application
return Ok(FinalCustomVisionResponse);
{
    // Make a Prediction object and set it to
    // the values in TempCustomVisionResponse.Predictions
    Prediction objPrediction = new Prediction();
    objPrediction.TagId = Prediction.TagId;
    objPrediction.TagName = Prediction.TagName;
    // Convert the Probability to a decimal
    Decimal dProbability = 0;
    Decimal.TryParse(Prediction.Probability, out dProbability);
    // Convert the decimal to a percentage
    objPrediction.Probability = dProbability.ToString("#0.##%");
    // Add the Prediction object to the Predictions
    FinalCustomVisionResponse.Predictions.Add(objPrediction);
}
```

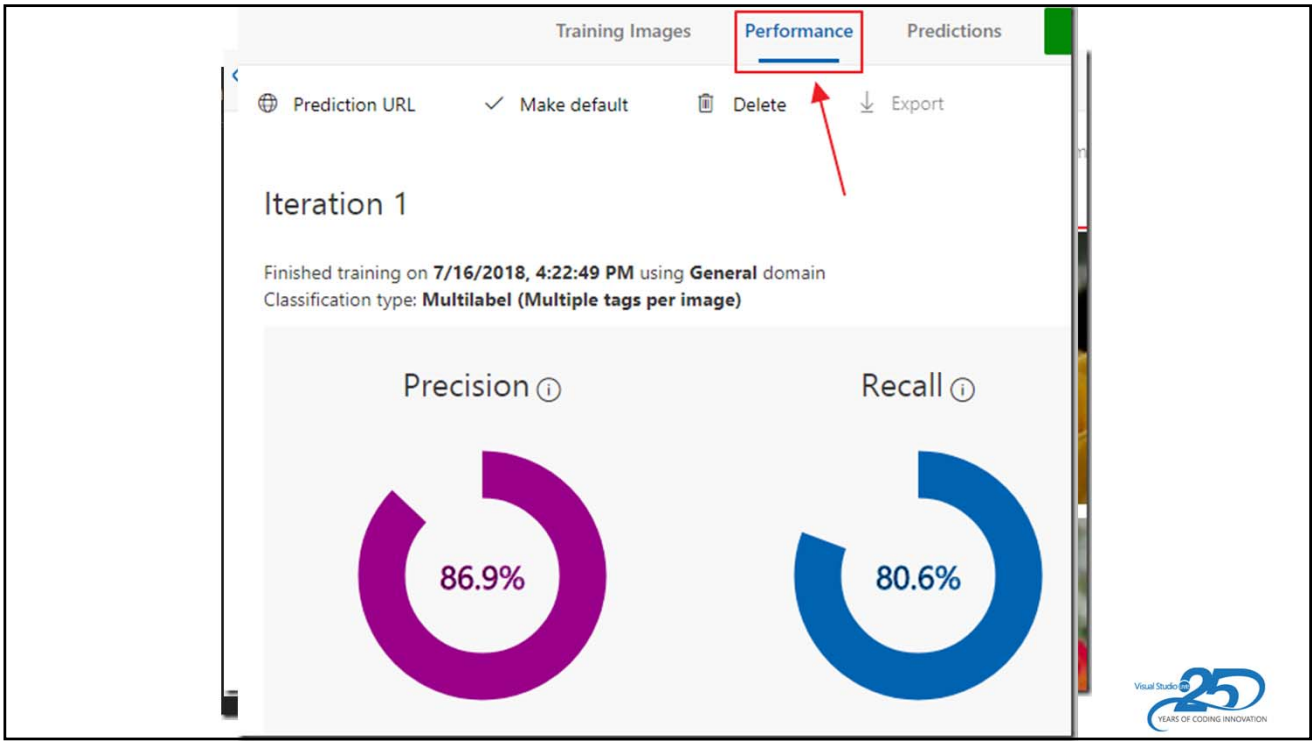
Angular Code




Demonstration


.Net Core Angular Application
(Part Two)


Train Project
Projects
Bracelets - Classify bracelet types
Tags
Men
Choose
Upload New Files
Cancel
List of Images
633ee8f9-4b93-4977-918b-b160a608d4b3
1ae06f8b-bfba-414a-9076-3357280f1573
5364294a-a90c-4f8d-bc37-e000b62cc4f6
133d4876-a9d2-46de-8431-2454391c0a9e
bc238cbb-e306-4be9-83e8-327ca9f296f4
3a72449c-8899-4c3f-80ef-ef16acab7d43
1240c4a2-b34c-4055-b41c-6d9c506c779f
b30e740e-1082-4183-8916-211ccb0757dd
1 2 3 4 5


Training Images
Performance
Predictions
Prediction URL
Make default
Delete
Export
Iteration 1
Finished training on 7/16/2018, 4:22:49 PM using General domain
Classification type: Multilabel (Multiple tags per image)
Precision
Recall
86.9%
80.6%

.Net Core Angular Application

(Create The Application)

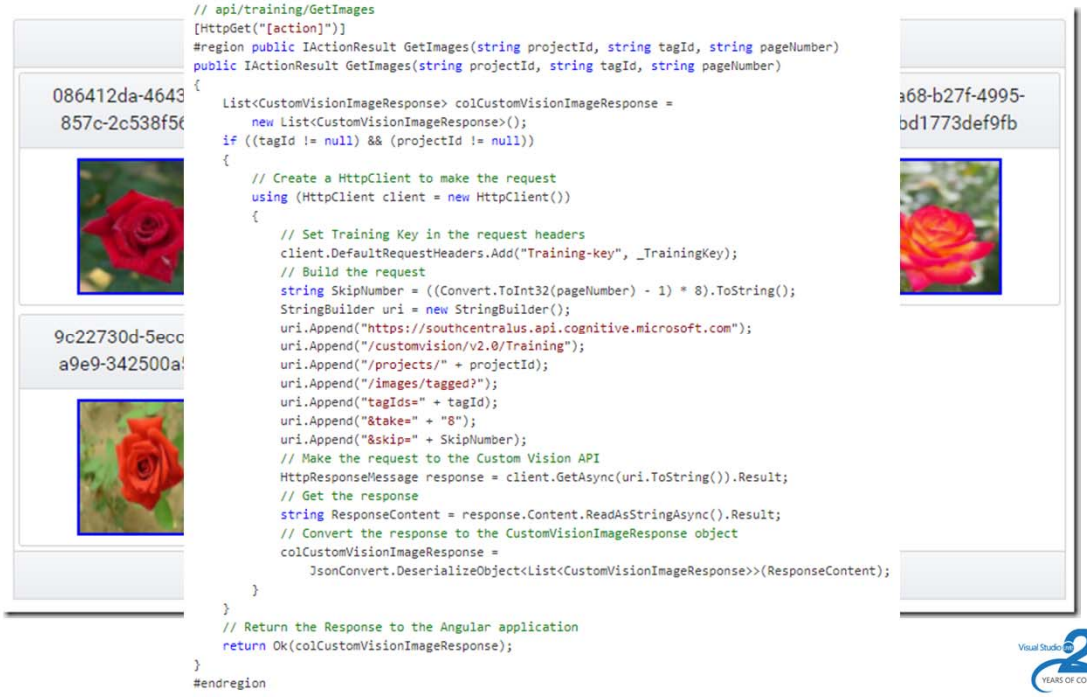
Get Projects

```
            public getProjects() {
// api/training/GetProjects
[HttpGet("[action]")]
#region public IActionResult GetProjects()
public IActionResult GetProjects()
{
    List<CustomVisionProjectResponse> colCustomVisionTrainingResponse =
        new List<CustomVisionProjectResponse>();
    // Create a HttpClient to make the request
    using (HttpClient client = new HttpClient())
    {
        // Set Training Key in the request headers
        client.DefaultRequestHeaders.Add("Training-key", _TrainingKey);
        // Build the request to the Custom Vision API
        StringBuilder uri = new StringBuilder();
        uri.Append("https://southcentralus.api.cognitive.microsoft.com");
        uri.Append("/customvision/v2.0/Training");
        uri.Append("/projects?");
        // Make the request to the Custom Vision API
        HttpResponseMessage response = client.GetAsync(uri.ToString()).Result;
        // Get the response
        string ResponseContent = response.Content.ReadAsStringAsync().Result;
        // Convert the response to the CustomVisionTrainingResponse object
        colCustomVisionTrainingResponse =
            JsonConvert.DeserializeObject<List<CustomVisionProjectResponse>>(ResponseContent);
    }
    // Return the Response to the Angular application
    return Ok(colCustomVisionTrainingResponse);
}
#endregion
            }
```

Get Tags

```
                public getTags() {
// api/training/GetTags
[HttpGet("[action]")]
#region public IActionResult GetTags(string projectId)
public IActionResult GetTags(string projectId)
{
    List<CustomVisionTagResponse> objCustomVisionTagResponse =
        new List<CustomVisionTagResponse>();
    // Create a HttpClient to make the request
    using (HttpClient client = new HttpClient())
    {
        // Set Training Key in the request headers
        client.DefaultRequestHeaders.Add("Training-key", _TrainingKey);
        // Build the request to the Custom Vision API
        StringBuilder uri = new StringBuilder();
        uri.Append("https://southcentralus.api.cognitive.microsoft.com");
        uri.Append("/customvision/v2.0/Training");
        uri.Append("/projects/" + projectId);
        uri.Append("/tags?");
        // Make the request to the Custom Vision API
        HttpResponseMessage response = client.GetAsync(uri.ToString()).Result;
        // Get the response
        string ResponseContent = response.Content.ReadAsStringAsync().Result;
        // Convert the response to the CustomVisionTagResponse object
        objCustomVisionTagResponse =
            JsonConvert.DeserializeObject<List<CustomVisionTagResponse>>(ResponseContent);
    }
    // Return the Response to the Angular application
    return Ok(objCustomVisionTagResponse);
}
#endregion
                ,
```

Get Images




```
// api/training/GetImages
[HttpGet("[action]")]
#region public IActionResult GetImages(string projectId, string tagId, string pageNumber)
public IActionResult GetImages(string projectId, string tagId, string pageNumber)
{
    List<CustomVisionImageResponse> colCustomVisionImageResponse =
        new List<CustomVisionImageResponse>();
    if ((tagId != null) && (projectId != null))
    {
        // Create a HttpClient to make the request
        using (HttpClient client = new HttpClient())
        {
            // Set Training Key in the request headers
            client.DefaultRequestHeaders.Add("Training-key", _TrainingKey);
            // Build the request
            string SkipNumber = ((Convert.ToInt32(pageNumber) - 1) * 8).ToString();
            StringBuilder uri = new StringBuilder();
            uri.Append("https://southcentralus.api.cognitive.microsoft.com");
            uri.Append("/customvision/v2.0/Training");
            uri.Append("/projects/" + projectId);
            uri.Append("/images/tagged?");
            uri.Append("tagIds=" + tagId);
            uri.Append("&take=" + "8");
            uri.Append("&skip=" + SkipNumber);
            // Make the request to the Custom Vision API
            HttpResponseMessage response = client.GetAsync(uri.ToString()).Result;
            // Get the response
            string ResponseContent = response.Content.ReadAsStringAsync().Result;
            // Convert the response to the CustomVisionImageResponse object
            colCustomVisionImageResponse =
                JsonConvert.DeserializeObject<List<CustomVisionImageResponse>>(ResponseContent);
        }
    }
    // Return the Response to the Angular application
    return Ok(colCustomVisionImageResponse);
}
#endregion
```

Uploading Images

```
if ((selectedProject != null) && (selectedTag != null))
{
    // Process all Files
    foreach (var file in form.Files)
    {
        // Process file
        using (var readStream = file.OpenReadStream())
        {
            // Create a HttpClient to make the request
            using (HttpClient client = new HttpClient())
            {
                // Set Training Key in the request headers
                client.DefaultRequestHeaders.Add("Training-key", _TrainingKey);
                // Serialize Request
                MultipartFormDataContent _multiPartContent =
                    new MultipartFormDataContent();
                StreamContent _imageData =
                    new StreamContent(readStream);
                _imageData.Headers.ContentType =
                    new MediaTypeHeaderValue("application/octet-stream");
                ContentDispositionHeaderValue _contentDispositionHeaderValue =
                    new ContentDispositionHeaderValue("form-data");
                _contentDispositionHeaderValue.Name = "imageData";
                _contentDispositionHeaderValue.FileName = file.Name;
                _imageData.Headers.ContentDisposition = _contentDispositionHeaderValue;
                _multiPartContent.Add(_imageData, "imageData");
                // Build the request to the Custom Vision API
                StringBuilder uri = new StringBuilder();
                uri.Append("https://southcentralus.api.cognitive.microsoft.com");
                uri.Append("/customvision/v2.0/Training");
                uri.Append("/projects/" + selectedProject);
                uri.Append("/images/image?");
                uri.Append("tagIds=" + selectedTag);
                // Make the request to the Custom Vision Service API
                HttpResponseMessage response =
                    client.PostAsync(uri.ToString(), _multiPartContent).Result;
                // Get the response -- (Do nothing with it for now)
                string ResponseContent = response.Content.ReadAsStringAsync().Result;
            }
        }
    }
}
```

Train Project

```
// api/training/TrainProject
[HttpGet("[action]")]
#region public IActionResult TrainProject(string projectId)
public IActionResult TrainProject(string projectId)
{
    CustomVisionTrainResponse objCustomVisionTrainResponse = new CustomVisionTrainResponse();
    // Create a HttpClient to make the request
    using (HttpClient client = new HttpClient())
    {
        HttpResponseMessage response;
        // Set Training Key in the request headers
        client.DefaultRequestHeaders.Add("Training-key", _TrainingKey);
        // Build the request to the Custom Vision API
        StringBuilder uri = new StringBuilder();
        uri.Append("https://southcentralus.api.cognitive.microsoft.com");
        uri.Append("/customvision/v2.0/Training");
        uri.Append("/projects/" + projectId);
        uri.Append("/train?");
        // Request body
        byte[] byteData = Encoding.UTF8.GetBytes("{body}");
        // Make the request to the Custom Vision Service API
        using (var content = new ByteArrayContent(byteData))
        {
            content.Headers.ContentType = new MediaTypeHeaderValue("application/json");
            response = client.PostAsync(uri.ToString(), content).Result;
        }
        if (response.StatusCode == System.Net.HttpStatusCode.BadRequest)
        {
            objCustomVisionTrainResponse.Status = "No Training needed";
        }
        else
        {
            // Get the response
            string ResponseContent = response.Content.ReadAsStringAsync().Result;
            // Convert the response to the CustomVisionTrainResponse object
            objCustomVisionTrainResponse =
                JsonConvert.DeserializeObject<CustomVisionTrainResponse>(ResponseContent);
        }
    }
    // Return a Response to the Angular application
    return Ok(objCustomVisionTrainResponse);
}
#endregion
```

Demonstration

Resources

AI Help Website
http://AIHelpWebsite.com

ADefWebserver
http://ADefWebserver.com

Questions ?

Thank You!