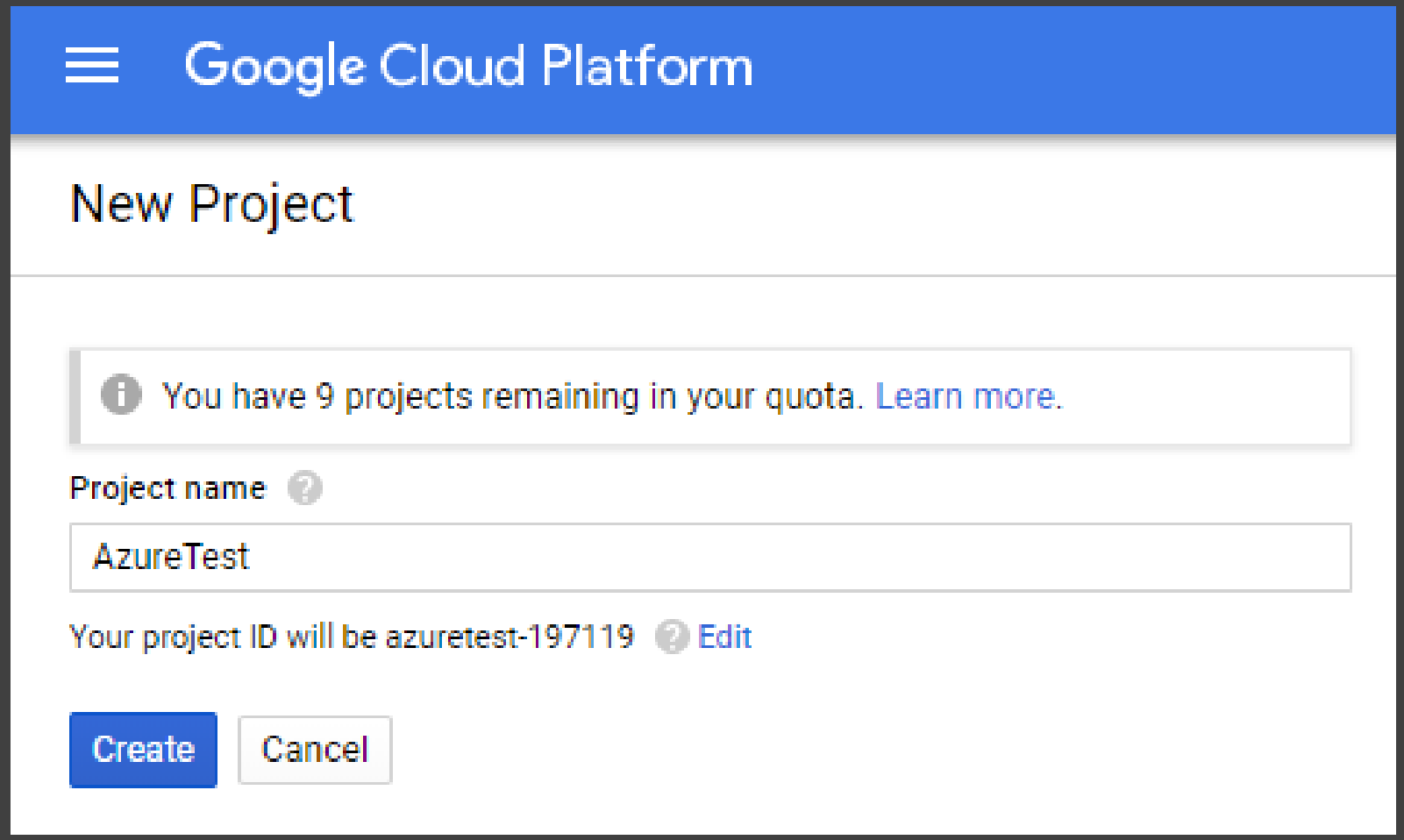
Microsoft Azure – Cloud Services

Milan Pekardy

pekardy@dcs.uni-pannon.hu

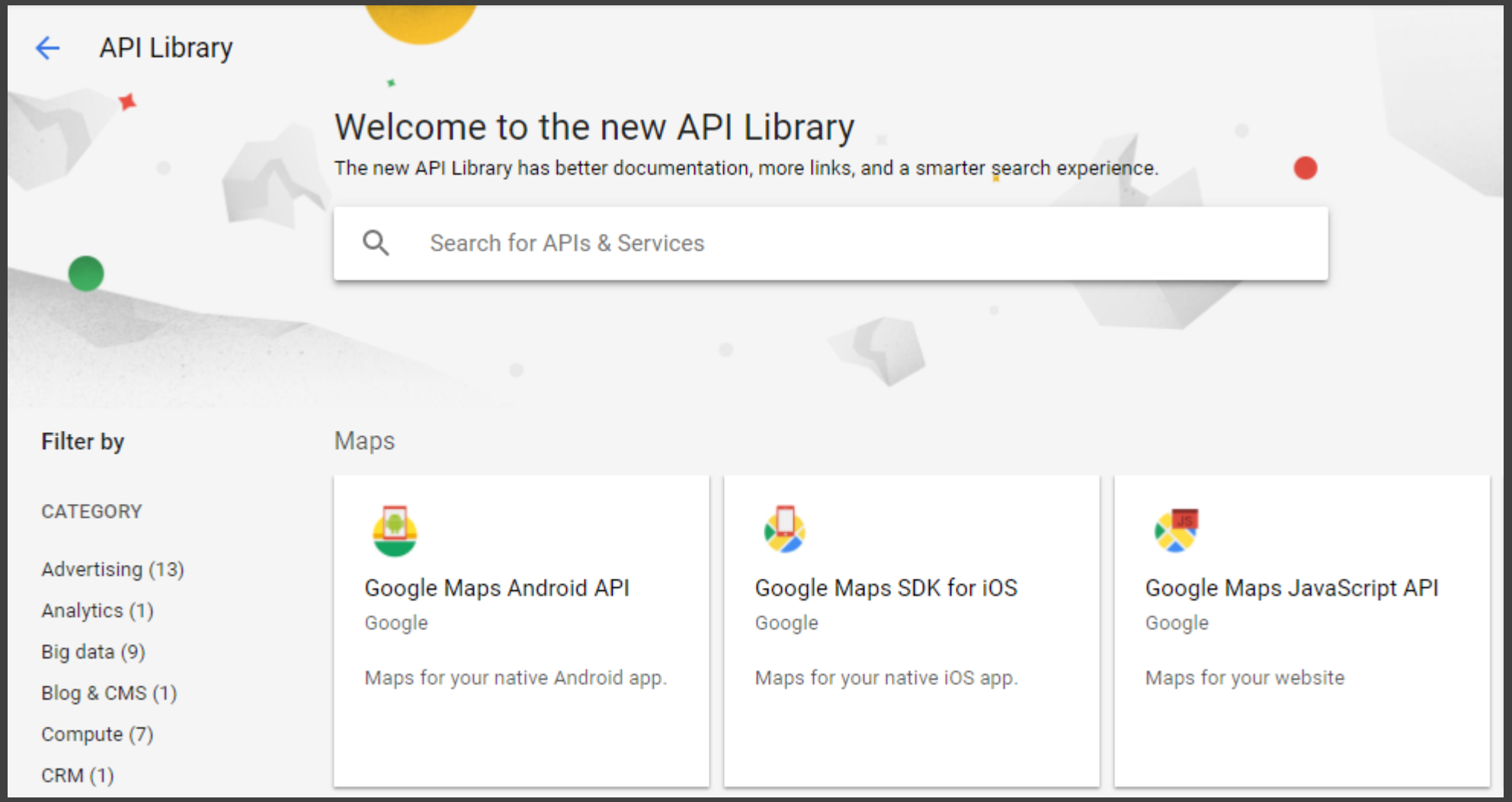
Google Cloud Platform

- Create a new project



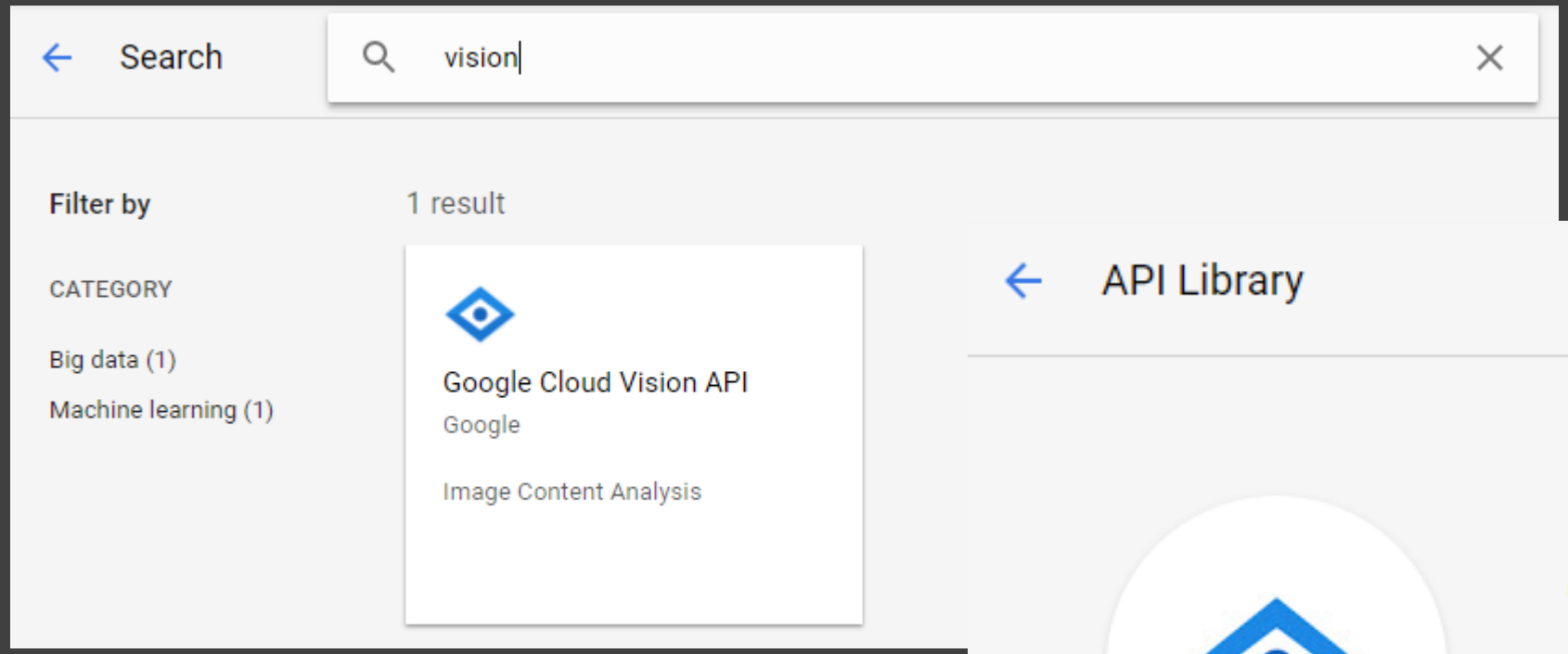
Google Cloud Platform

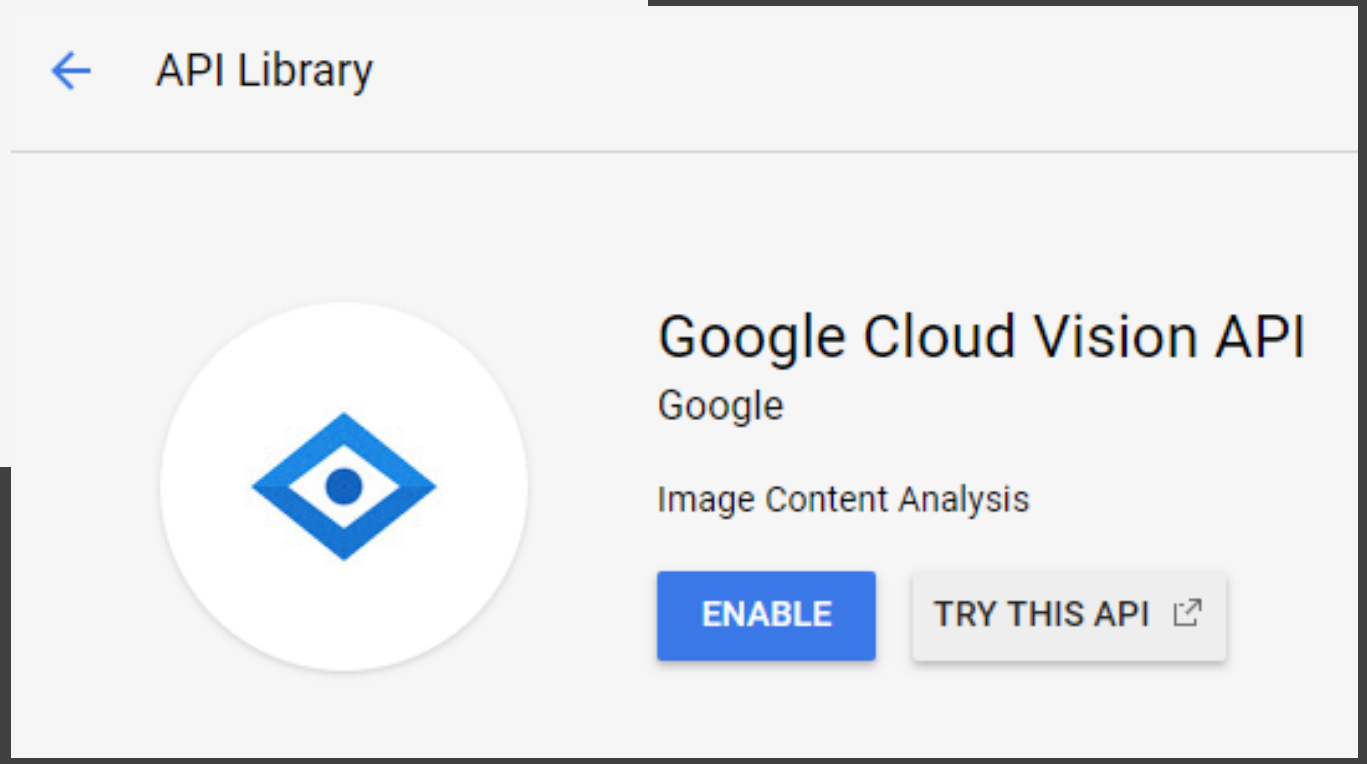
- Select the APIs you want to use from the API library



Exercise 1 – Google Cloud Vision API

- Use the vision API to detect labels in Todo pictures

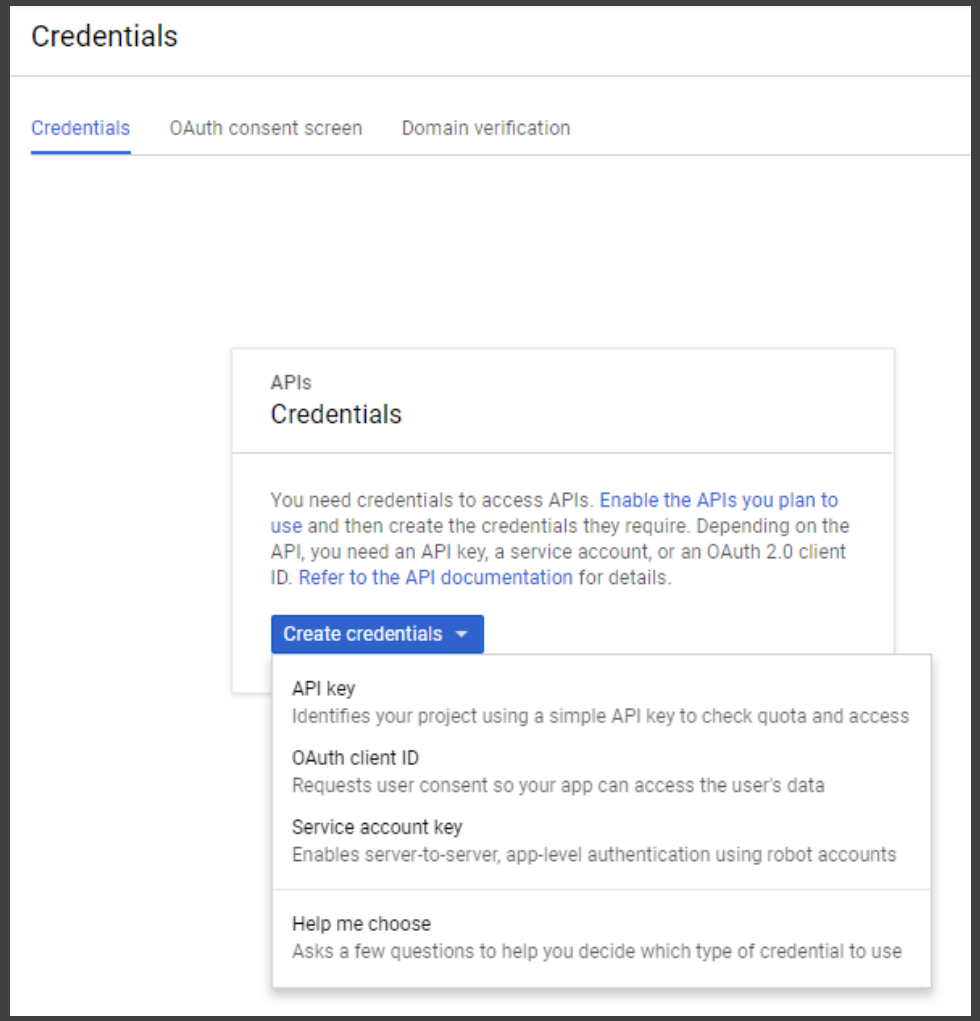


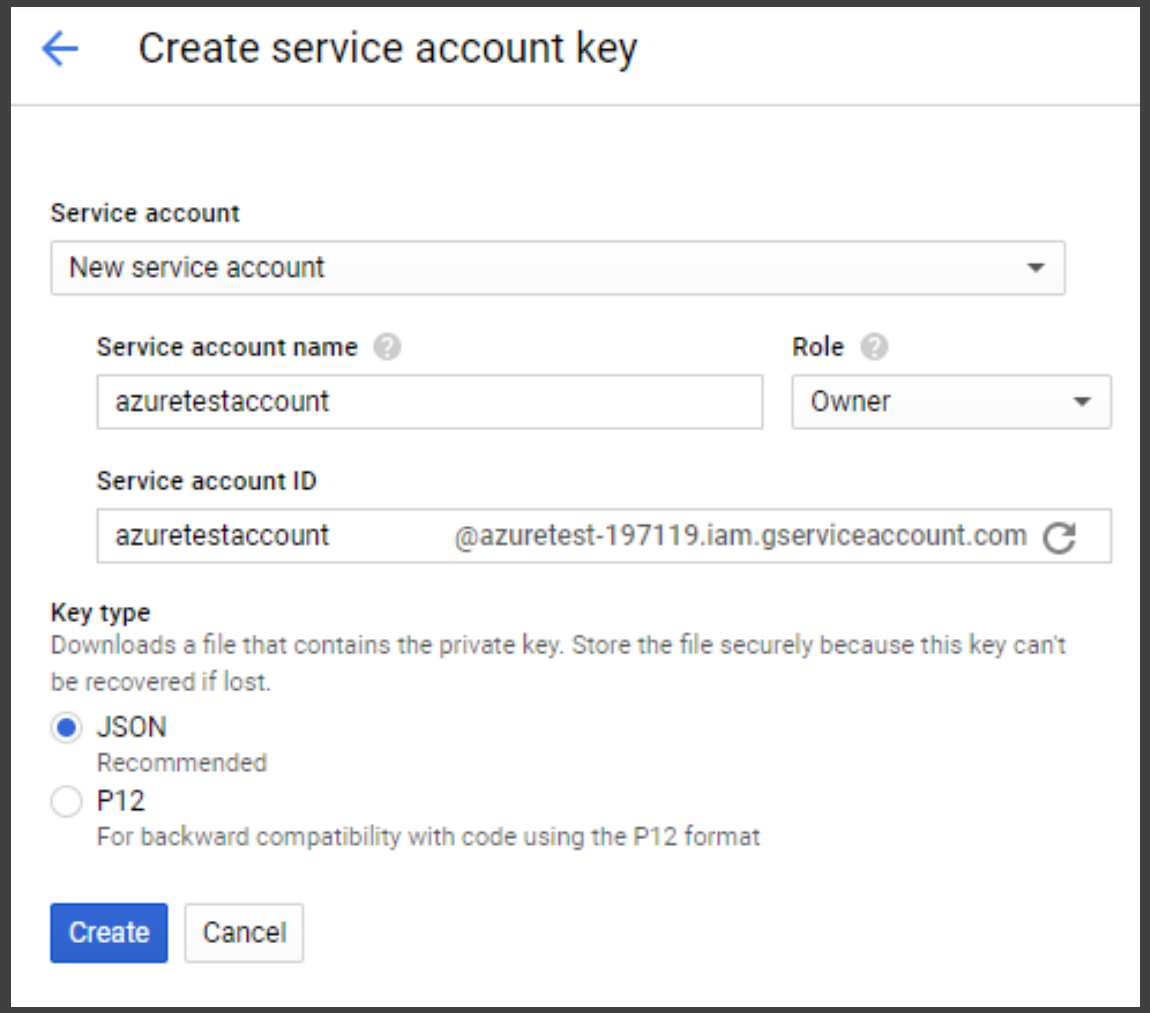


- You need to add a billing account

Exercise 1 – Google Cloud Vision API

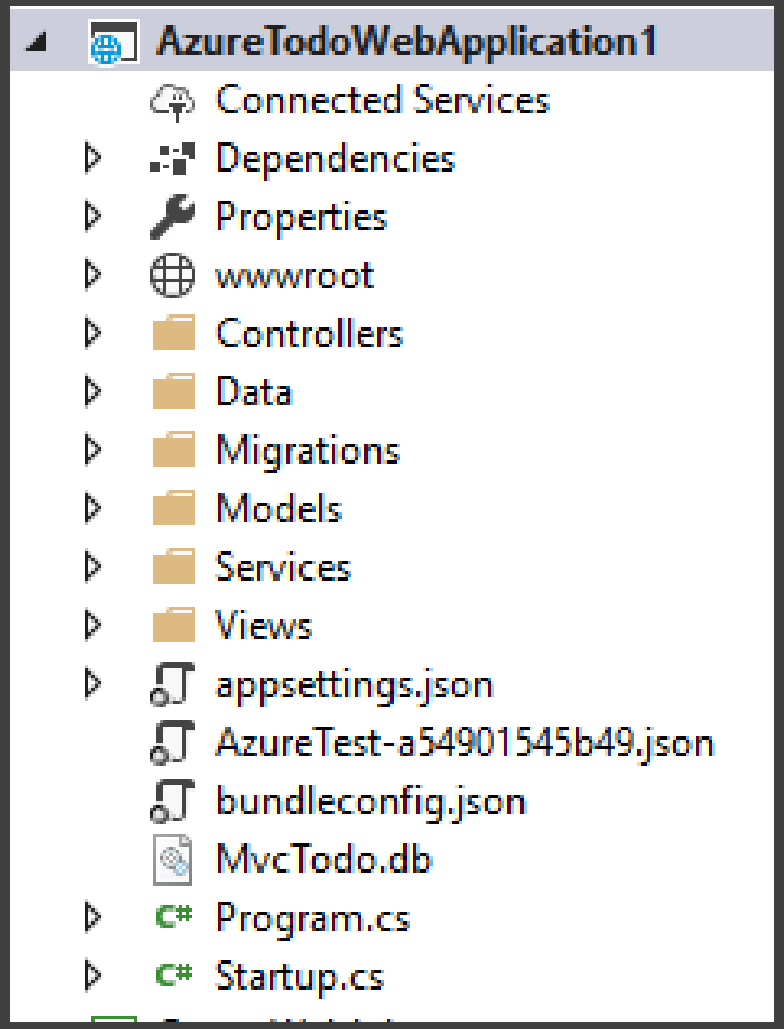
- To consume the API create the appropriate credentials
  - APIs & Services > Credentials > Create credentials > Service account key





Exercise 1 – Google Cloud Vision API

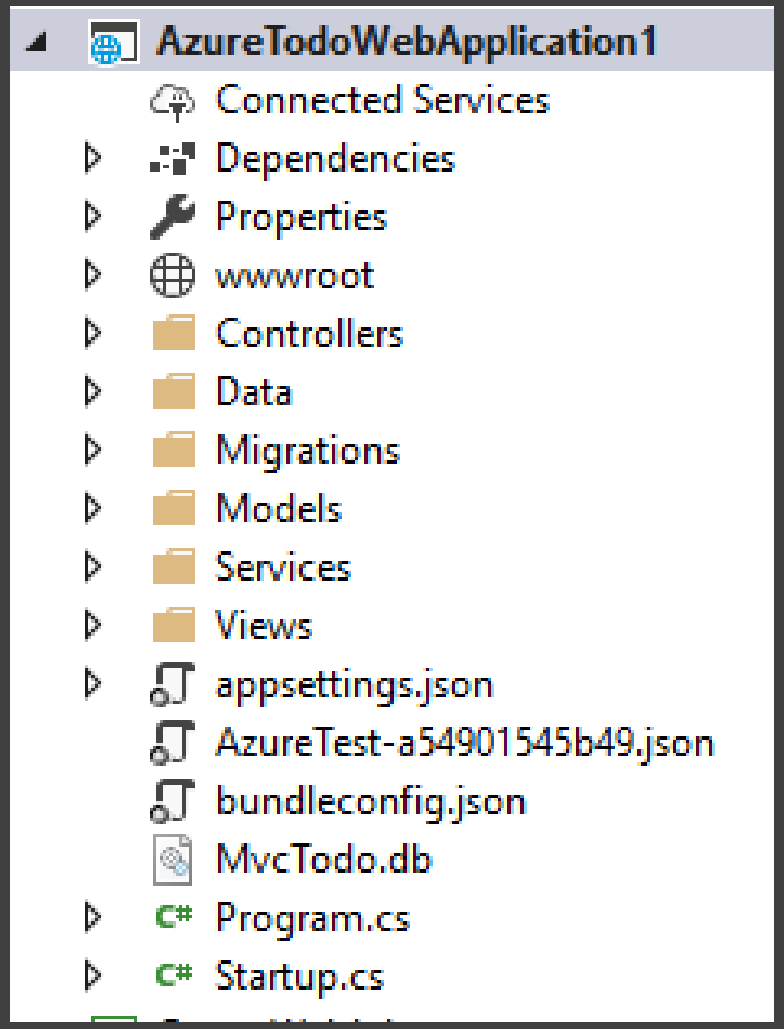
- Download the credentials JSON file and place it in your web app's root folder



Exercise 1 – Google Cloud Vision API

- Download the credentials JSON file and place it in your web app's root folder
- Create a service interface

```
public interface IGoogleCloudService
{
    2 references | 0 exceptions
    Task<string> GetImageLabelsAsync(string imageUri);
}
```



Exercise 1 – Google Cloud Vision API

- Implement the interface

```
public class GoogleCloudService : IGoogleCloudService
{
    private readonly IHostingEnvironment _hostingEnvironment;

    0 references | 0 exceptions
    public GoogleCloudService(IHostingEnvironment hostingEnvironment)
    {
        _hostingEnvironment = hostingEnvironment;
    }

    2 references | 0 exceptions
    public async Task<string> GetImageLabelsAsync(string imageUri)
    {
        string path = Path.Combine(_hostingEnvironment.ContentRootPath, "AzureTest-a54901545b49.json");
        GoogleCredential credential = GoogleCredential.FromFile(path).CreateScoped(ImageAnnotatorClient.DefaultScopes);
        Channel channel = new Channel(ImageAnnotatorClient.DefaultEndpoint.ToString(), credential.ToChannelCredentials());

        using (WebClient wc = new WebClient())
        {
            Image image = Image.FromBytes(await wc.DownloadDataTaskAsync(imageUri));
            ImageAnnotatorClient client = ImageAnnotatorClient.Create(channel);
            IReadOnlyList<EntityAnnotation> labels = await client.DetectLabelsAsync(image);
            string labelsString = "";
            foreach (EntityAnnotation label in labels)
            {
                labelsString += label.Description + "|";
            }
            return labelsString;
        }
    }
}
```

Exercise 1 – Google Cloud Vision API

- Extend the Todo model to store the labels

```
public class Todo
{
    [Key]
    14 references | 0 exceptions
    public Guid Id { get; set; }

    [Required(AllowEmptyStrings = false, ErrorMessage = "Todo name is required!")]
    [StringLength(100, ErrorMessage = "Todo name max. length is 100!")]
    8 references | 0 exceptions
    public string Name { get; set; }

    [StringLength(1000, ErrorMessage = "Todo description max. length is 1000!")]
    7 references | 0 exceptions
    public string Description { get; set; }

    [Display(Name = "Created Date")]
    [DataType(DataType.Date)]
    [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
    7 references | 0 exceptions
    public DateTime CreatedDate { get; set; }

    [StringLength(200, ErrorMessage = "PhotoUrl max. length is 200!")]
    5 references | 0 exceptions
    public string PhotoUrl { get; set; }
    [NotMapped]
    2 references | 0 exceptions
    public string PhotoLabels { get; set; }
}
```

Exercise 1 – Google Cloud Vision API

- Extend the Details controller action to populate the labels with the service response

```
if(todo.PhotoUrl != null)
{
    todo.PhotoLabels = await _googleCloudService.GetImageLabelsAsync(todo.PhotoUrl);
}
```

- Extend the Details view to show the picture labels

```
@if (Model.PhotoUrl != null)
{
    <dd>
        <img src="@Model.PhotoUrl" title="@Model.Name" />
    </dd>
    <dt>
        Photo labels:
    </dt>
    <dd>
        @Html.DisplayFor(model => model.PhotoLabels)
    </dd>
}
```

Exercise 2 – Google Cloud Vision API

- Modify the label detection so that the service call only executed once per picture:
  - Map the entity attribute to the database (migration),
  - In the Details controller action call the service only if the labels field not filled yet