# VkDescriptorSetLayout

```
typedef enum VkDescriptorType {
    VK_DESCRIPTOR_TYPE_SAMPLER = 0,
    VK_DESCRIPTOR_TYPE_COMBINED_IMAGE_SAMPLER = 1,
    VK_DESCRIPTOR_TYPE_SAMPLED_IMAGE = 2,
    VK_DESCRIPTOR_TYPE_STORAGE_IMAGE = 3,
    VK_DESCRIPTOR_TYPE_UNIFORM_TEXEL_BUFFER = 4,
    VK_DESCRIPTOR_TYPE_STORAGE_TEXEL_BUFFER = 5,
    VK_DESCRIPTOR_TYPE_UNIFORM_BUFFER = 6,
    VK_DESCRIPTOR_TYPE_STORAGE_BUFFER = 7,
    VK_DESCRIPTOR_TYPE_UNIFORM_BUFFER_DYNAMIC = 8,
    VK_DESCRIPTOR_TYPE_STORAGE_BUFFER_DYNAMIC = 9,
    VK_DESCRIPTOR_TYPE_INPUT_ATTACHMENT = 10,
...
} VkDescriptorType;
```

```
typedef enum VkShaderStageFlagBits {
    VK_SHADER_STAGE_VERTEX_BIT = 0x00000001,
    VK_SHADER_STAGE_TESSELLATION_CONTROL_BIT = 0x00000002,
    VK_SHADER_STAGE_TESSELLATION_EVALUATION_BIT = 0x00000004,
    VK_SHADER_STAGE_GEOMETRY_BIT = 0x00000008,
    VK_SHADER_STAGE_FRAGMENT_BIT = 0x00000010,
    VK_SHADER_STAGE_COMPUTE_BIT = 0x00000020,
    VK_SHADER_STAGE_ALL_GRAPHICS = 0x0000001F,
    VK_SHADER_STAGE_ALL = 0x7FFFFFFF,
...
} VkShaderStageFlagBits;
```

```
typedef struct VkDescriptorSetLayoutBinding {
    uint32_t              binding;
        // This must match the binding number in the shaders.
        // Ex. layout(binding = 0) uniform UniformBufferObject{...}ubo;
        //     layout(binding = 1) uniform sampler2D texSampler;

    VkDescriptorType      descriptorType;

    uint32_t              descriptorCount;
        // number of values in the array

    VkShaderStageFlags    stageFlags;

    const VkSampler*      pImmutableSamplers; // usually nullptr
} VkDescriptorSetLayoutBinding;
```

```
VkDescriptorSetLayoutCreateInfo {
sType = VK_STRUCTURE_TYPE_DESCRIPTOR_SET_LAYOUT_CREATE_INFO;
pNext = nullptr;
flags; // usually 0
bindingCount;
pBindings;
```

```
VkResult vkCreateDescriptorSetLayout(
    VkDevice                                    device,
    const VkDescriptorSetLayoutCreateInfo*      pCreateInfo,
    const VkAllocationCallbacks*                pAllocator,
    VkDescriptorSetLayout*                      pSetLayout
);
```

VkDevice

VkDescriptorSetLayout

```
void vkDestroyDescriptorSetLayout(
    VkDevice                    device,
    VkDescriptorSetLayout       descriptorSetLayout,
    const VkAllocationCallbacks* pAllocator);
```

VkDevice

VkDescriptorSetLayout