VkImageView

```
VkImageViewCreateInfo
sType = VK_STRUCTURE_TYPE_IMAGE_VIEW_CREATE_INFO;;
pNext = // usually nullptr;
flags = // usually 0;
image;                                   VkImage
viewType // VK_IMAGE_VIEW_TYPE_2D;
format; // Ex. VK_FORMAT_B8G8R8A8_SRGB

// swizzling is like var.[xyzw] in shader language
components.r = VK_COMPONENT_SWIZZLE_IDENTITY;
    // VK_COMPONENT_SWIZZLE_R

components.g = VK_COMPONENT_SWIZZLE_IDENTITY;
    // VK_COMPONENT_SWIZZLE_G

components.b = VK_COMPONENT_SWIZZLE_IDENTITY;
    // VK_COMPONENT_SWIZZLE_B

components.a = VK_COMPONENT_SWIZZLE_IDENTITY;
    // VK_COMPONENT_SWIZZLE_A

subresourceRange.aspectMask =VK_IMAGE_ASPECT_COLOR_BIT;
subresourceRange.baseMipLevel = 0;
subresourceRange.levelCount = 1;
subresourceRange.baseArrayLayer = 1;
subresourceRange.layerCount = 1;
```

```
typedef enum VkFormat {
    ...
    VK_FORMAT_R8G8B8_UNORM = 23,
    ...
    VK_FORMAT_R8G8B8_SRGB = 29,
    ...
} VkFormat;
```

```
enum VkImageAspectFlagBits
VK_IMAGE_ASPECT_COLOR_BIT = 0x00000001,
VK_IMAGE_ASPECT_DEPTH_BIT = 0x00000002,
VK_IMAGE_ASPECT_STENCIL_BIT = 0x00000004,
VK_IMAGE_ASPECT_METADATA_BIT = 0x00000008,
VK_IMAGE_ASPECT_PLANE_0_BIT = 0x00000010,
VK_IMAGE_ASPECT_PLANE_1_BIT = 0x00000020,
VK_IMAGE_ASPECT_PLANE_2_BIT = 0x00000040,
VK_IMAGE_ASPECT_NONE = 0,
VK_IMAGE_ASPECT_MEMORY_PLANE_0_BIT_EXT = 0x00000080,
VK_IMAGE_ASPECT_MEMORY_PLANE_1_BIT_EXT = 0x00000100,
VK_IMAGE_ASPECT_MEMORY_PLANE_2_BIT_EXT = 0x00000200,
VK_IMAGE_ASPECT_MEMORY_PLANE_3_BIT_EXT = 0x00000400,
VK_IMAGE_ASPECT_PLANE_0_BIT_KHR = VK_IMAGE_ASPECT_PLANE_0_BIT,
VK_IMAGE_ASPECT_PLANE_1_BIT_KHR = VK_IMAGE_ASPECT_PLANE_1_BIT,
VK_IMAGE_ASPECT_PLANE_2_BIT_KHR = VK_IMAGE_ASPECT_PLANE_2_BIT,
VK_IMAGE_ASPECT_NONE_KHR = VK_IMAGE_ASPECT_NONE,
```

```
VkResult vkCreateImageView(
    VkDevice                    device,          VkDevice
    const VkImageViewCreateInfo* pCreateInfo,
    const VkAllocationCallbacks* pAllocator,
    VkImageView*                pView
);
```

```
void vkDestroyImageView(
    VkDevice                    device,          VkDevice
    VkImageView                 imageView,       VkImageView
    const VkAllocationCallbacks* pAllocator
);
```