Есть несколько способов хранения изображений в API-приложении и отправки их по запросу к ASP.NET приложению.  
  
Один из способов - это хранение изображений в файловой системе API-приложения, и отправка пути к файлу изображения по запросу к ASP.NET приложению. Вот пример:  
  
1. В API-приложении, сохраните изображение на сервере, например, в папку "Images" сгруппированную по ID пользователя или другому идентификатору:  
  
```csharp  
// Получаем расширение файла  
string fileExtension = Path.GetExtension(imageFile.FileName);  
// Генерируем уникальное имя файла  
string uniqueFileName = Guid.NewGuid().ToString() + fileExtension;  
// Сохраняем изображение по указанному пути  
string filePath = Path.Combine("Images", uniqueFileName);  
using (var stream = new FileStream(filePath, FileMode.Create))  
{  
await imageFile.CopyToAsync(stream);  
}  
// Возвращаем путь к сохраненному файлу  
return Ok(filePath);  
```  
  
2. В вашем ASP.NET приложении, создайте метод, который будет получать путь к файлу изображения от API-приложения, например:  
  
```csharp  
public ActionResult GetImage(string filePath)  
{  
// Читаем бинарные данные из файла  
byte[] imageBytes = System.IO.File.ReadAllBytes(filePath);  
// Возвращаем файл изображения в ответе  
return File(imageBytes, "image/jpeg");  
}  
```  
  
Затем, вы можете отправить запрос из ASP.NET приложения к API-приложению, чтобы получить путь к файлу изображения и отобразить изображение на вашей ASP.NET странице. Например:  
  
```csharp  
public ActionResult ShowImage()  
{  
// Отправляем GET-запрос к API для получения пути к файлу изображения  
HttpClient client = new HttpClient();  
HttpResponseMessage response = await client.GetAsync("https://your-api-url.com/getimage");  
if (response.IsSuccessStatusCode)  
{  
// Получаем путь к файлу изображения из ответа  
string filePath = await response.Content.ReadAsStringAsync();  
// Отправляем запрос к ASP.NET приложению для получения самого изображения  
return RedirectToAction("GetImage", new { filePath = filePath });  
}  
else  
{  
// Обработка ошибки  
return RedirectToAction("Error");  
}  
}  
```  
  
Однако, у этого подхода есть свои ограничения, например, необходимость обращения к API для получения изображения и возможность конфликта имени файлов при сохранении.  
  
Другой способ - это использование специализированных сервисов хранения файлов, таких как Amazon S3, Azure Blob Storage или Google Cloud Storage. В этом случае, вам придется настроить соответствующий сервис хранения файлов и использовать соответствующие SDK или API для загрузки и получения файлов из облачного хранилища.