Использование Swagger/OpenAPI в ASP.NET

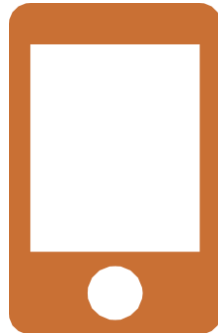
Алексей Кураколов
Архитектор «Loymax»
a.kourakolov@loymax.ru

Проблемы

Web

Back End

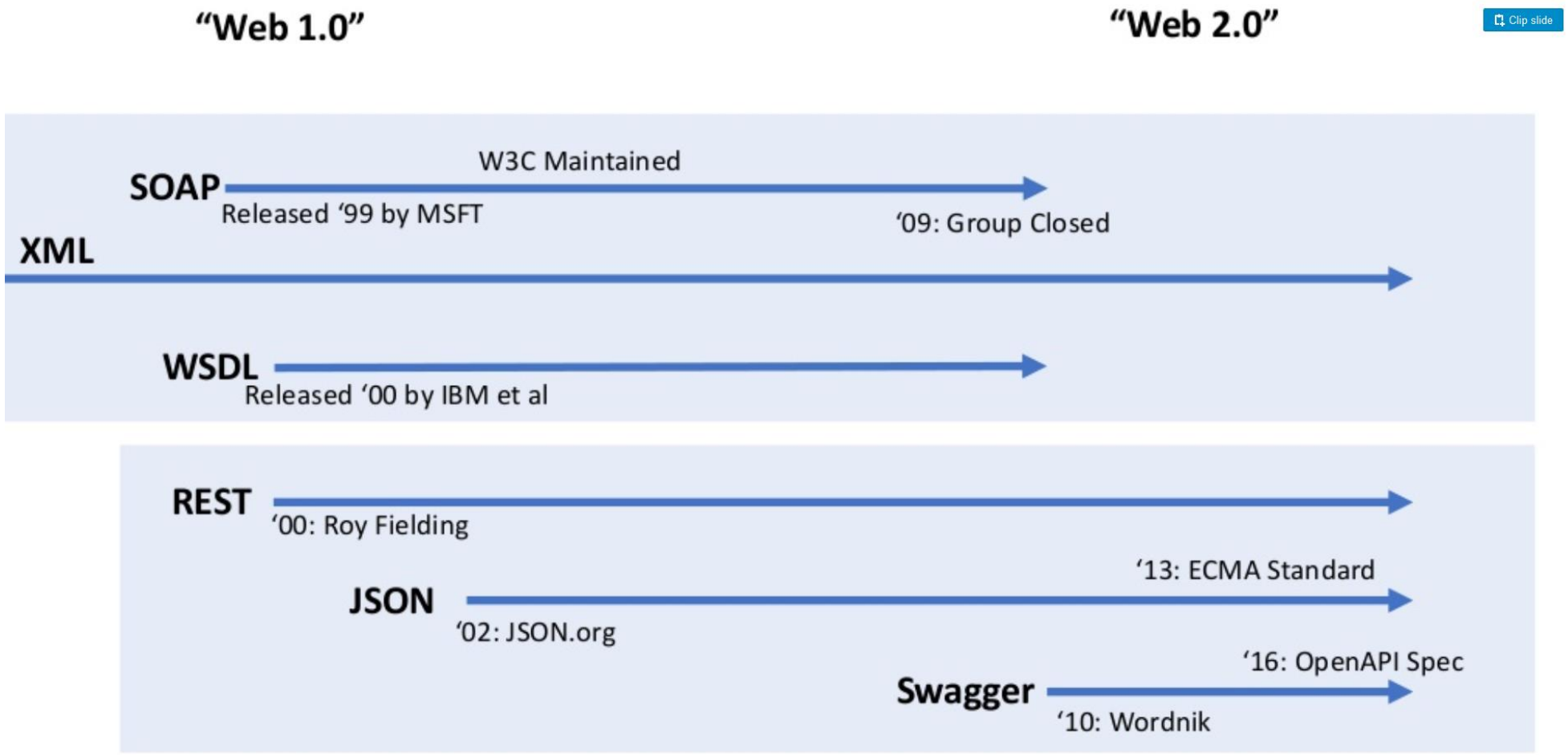
API



Mobile

Third party tools

История


“Web 1.0”
“Web 2.0”
Clip slide
W3C Maintained
SOAP
Released ‘99 by MSFT
‘09: Group Closed
XML
WSDL
Released ‘00 by IBM et al
REST
‘00: Roy Fielding
‘13: ECMA Standard
JSON
‘02: JSON.org
‘16: OpenAPI Spec
Swagger
‘10: Wordnik

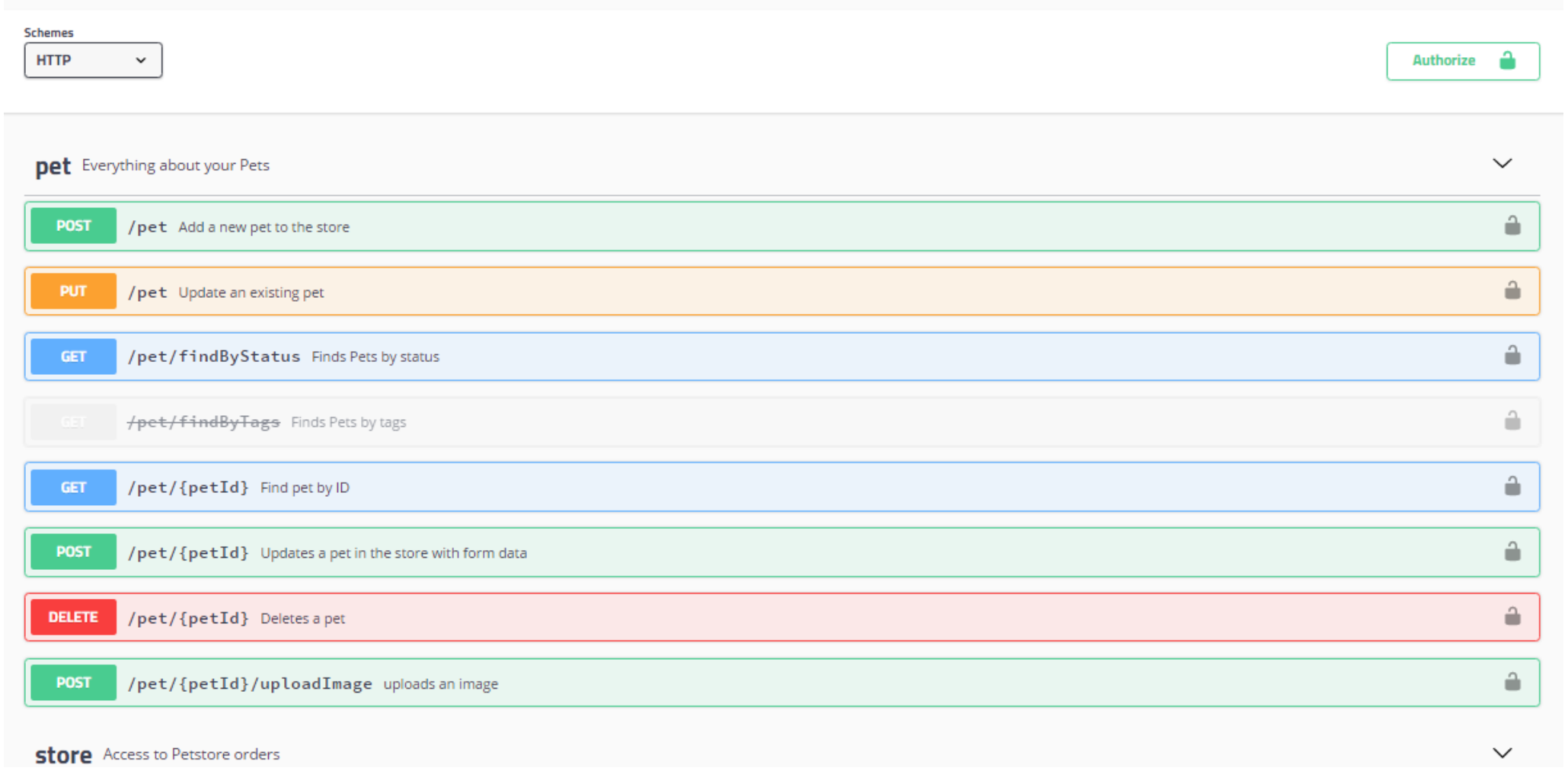
Задачи

1. Сделать API наглядным, понятным
2. Написать документацию по API
3. Облегчить написание клиентов к API, а лучше SDK
4. Версионирование API

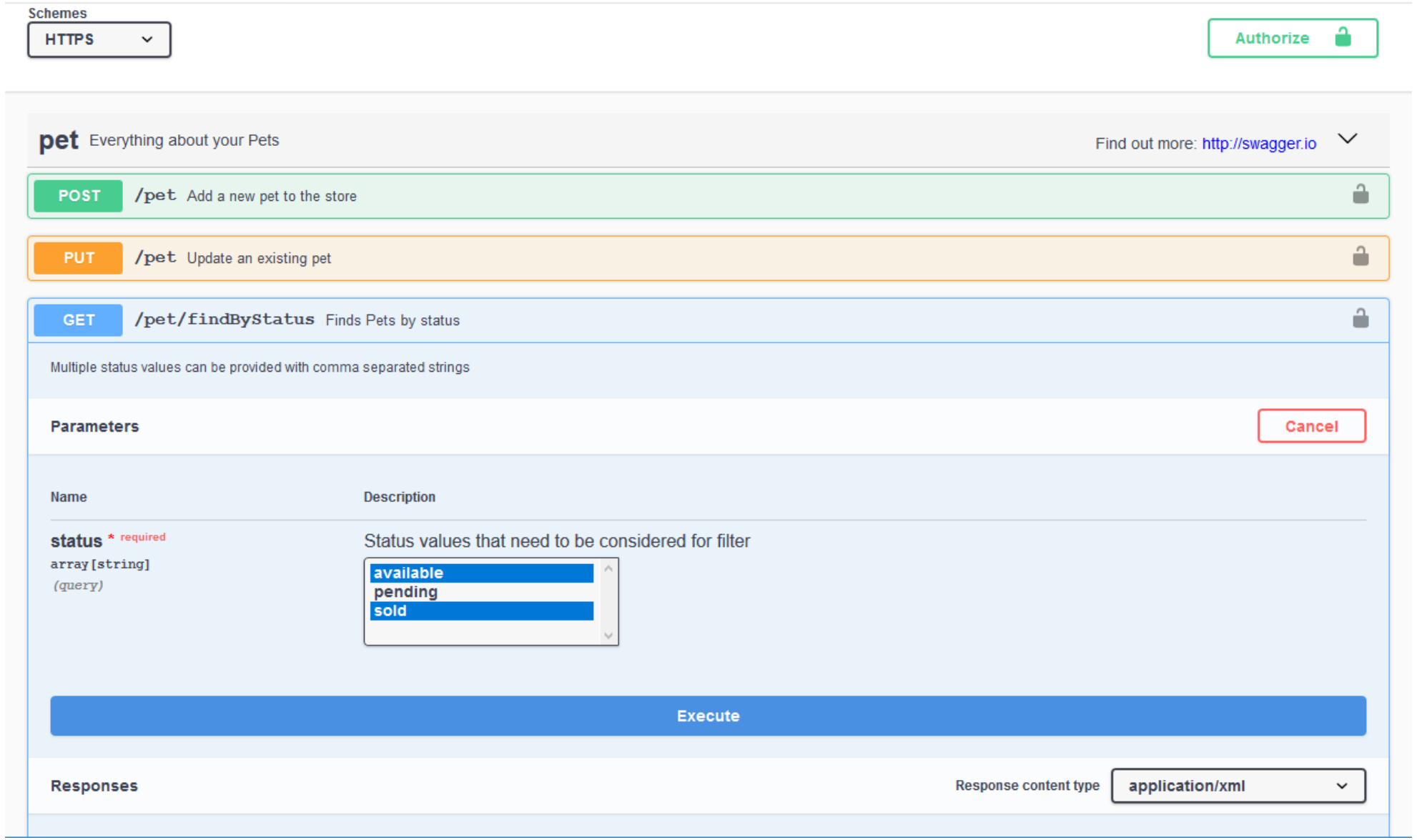
Почему OpenAPI?

1. Наглядность
2. Открытость и расширяемость
3. Большое сообщество и эко-система
4. Создано множество сторонних инструментов и сервисов

Почему OpenAPI?

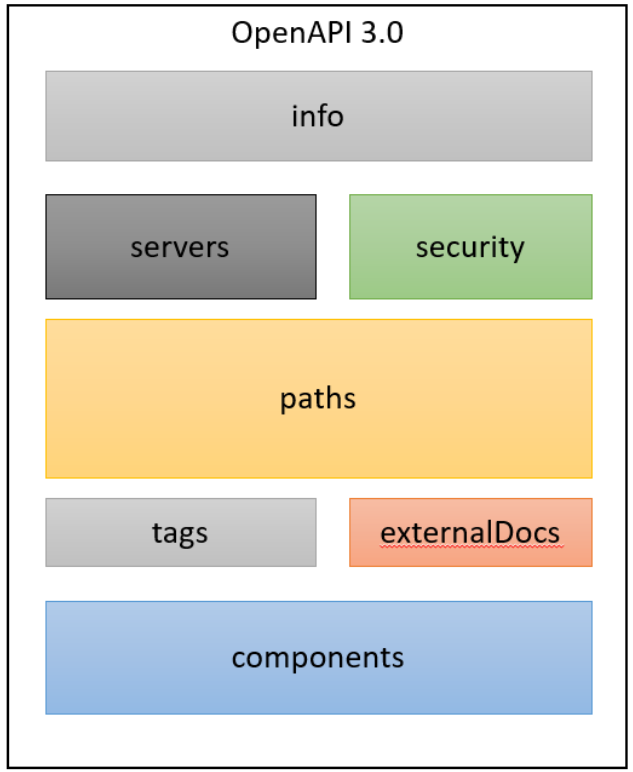


Почему OpenAPI?



Структура спецификации

Paths и Operations

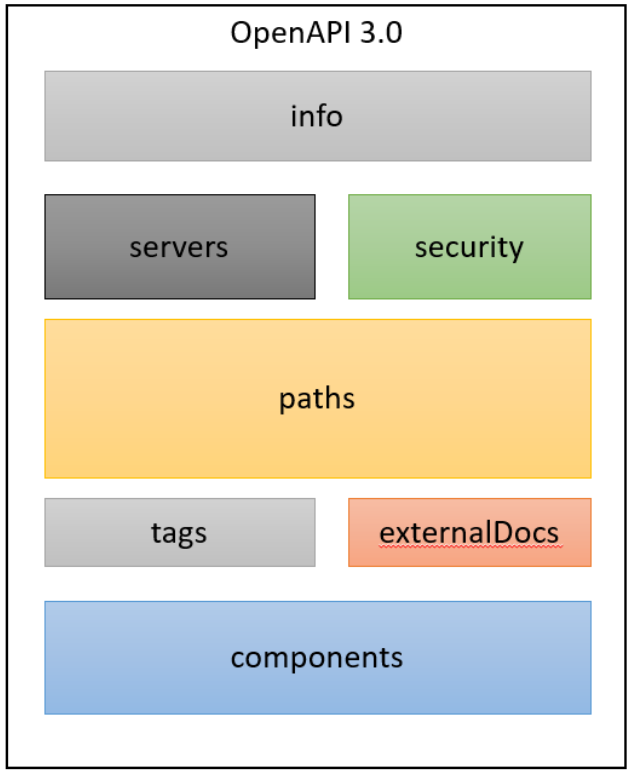


Paths – это ресурсы api/v1/books и api/v1/reports
Operations – методы HTTP для работы с ресурсами: get; post и т.д.

```yaml
paths:
  /users/{id}:
    summary: Represents a user
    description: >
      This resource represents an individual user in the system.
      Each user is identified by a numeric `id`.

    get:
      ...
    patch:
      ...
    delete:
      ...
```

Структура спецификации

Operations parameters

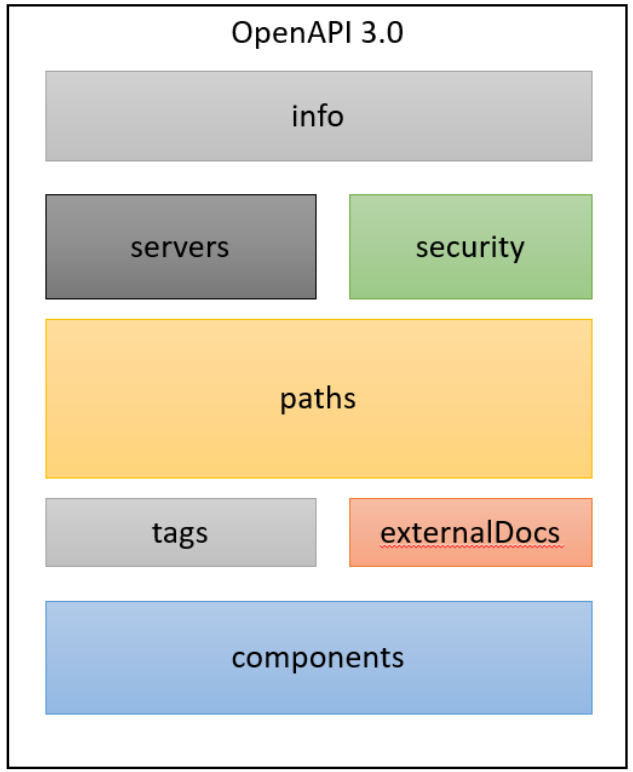


Параметры могут передаваться в path, query string, headers, body и cookie.

```yaml
paths:
  /users:
    get:
      parameters:
        - in: query
          name: role
          schema:
            type: string
            enum: [user, poweruser, admin]
          required: true
```

Структура спецификации

Data models

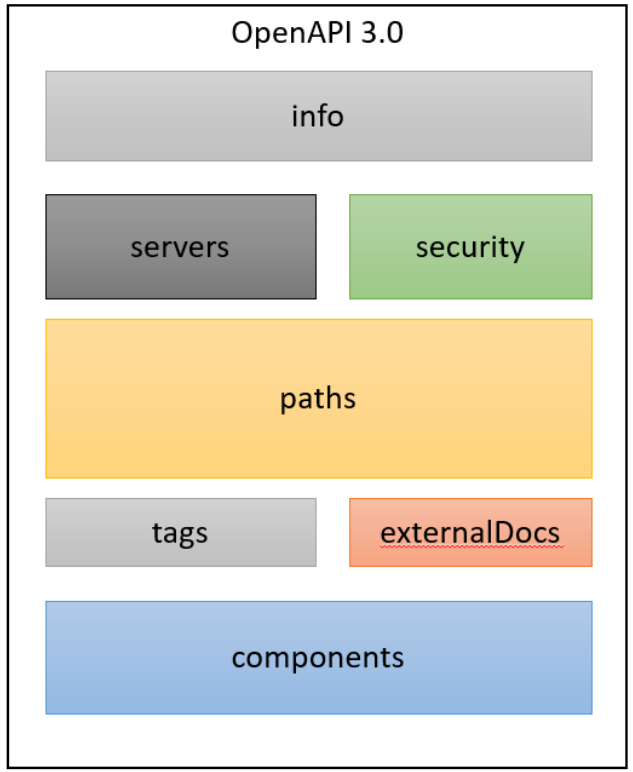


Базовые типы, массивы, полиморфизм и наследование.

```yaml
paths:
  /users:
    get:
      parameters:
        - in: query
          name: role
          schema:
            type: string
            enum: [user, poweruser, admin]
          required: true
```

Структура спецификации

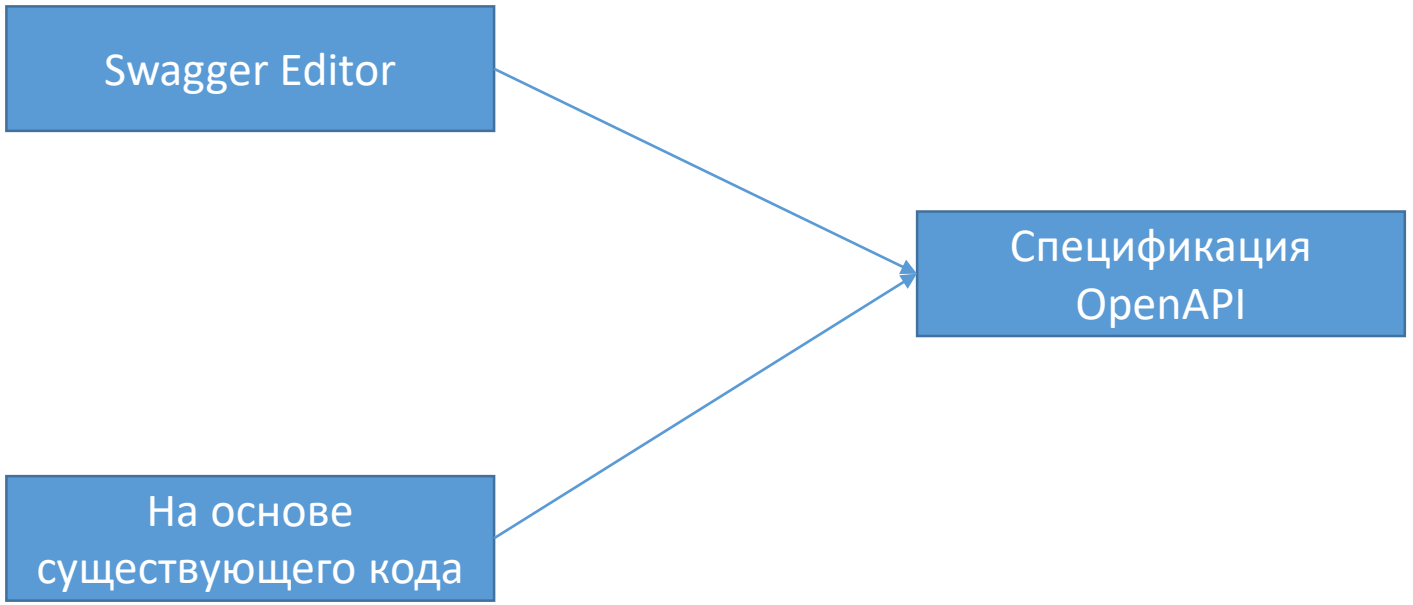
Security



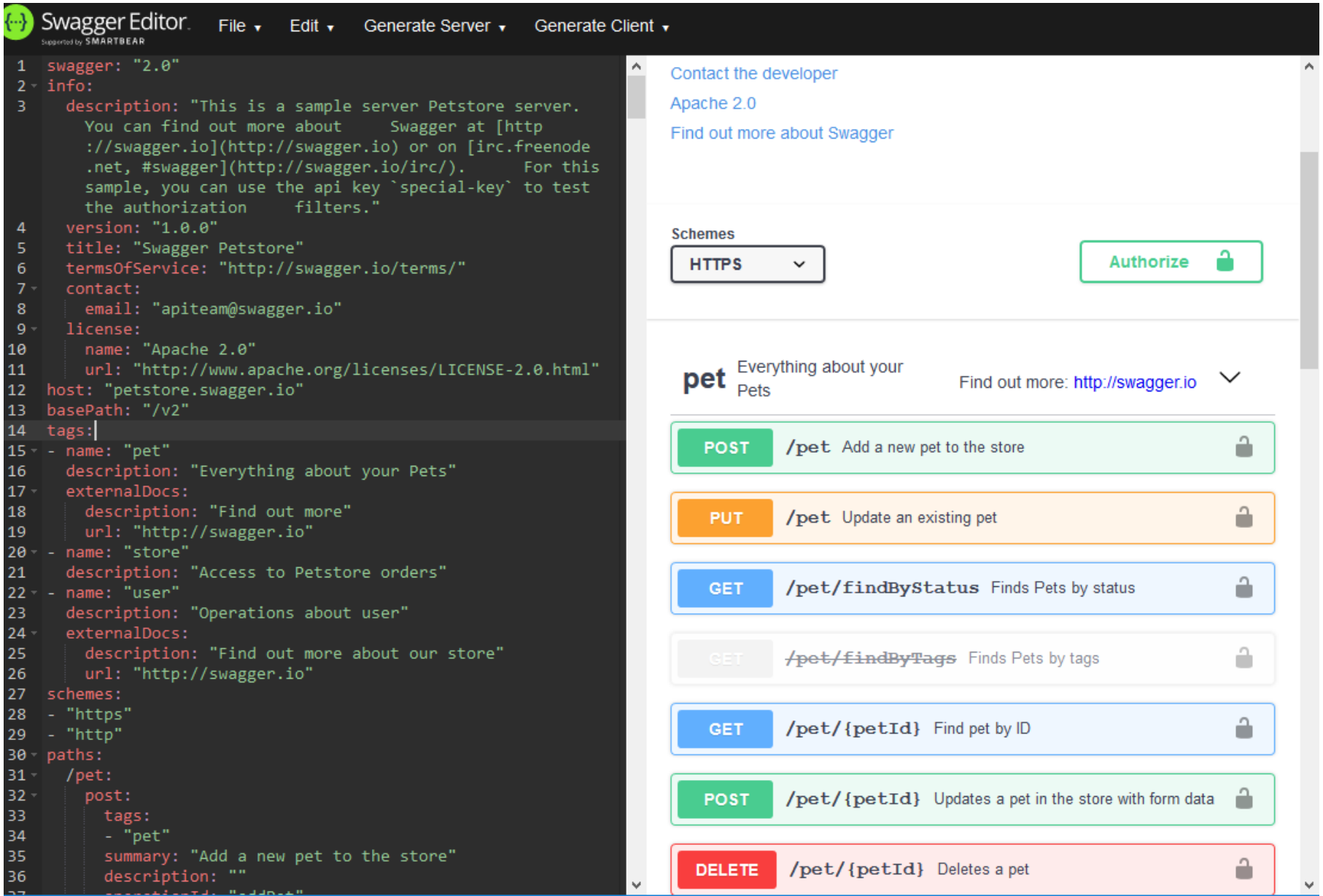
Описываются несколько способов авторизации.
Каждому path/operations можно задать свой набор совместимых авторизаций

```yaml
components:
  securitySchemes:

    BasicAuth:
      type: http
      scheme: basic

    BearerAuth:
      type: http
      scheme: bearer

    ApiKeyAuth:
      type: apiKey
      in: header
```

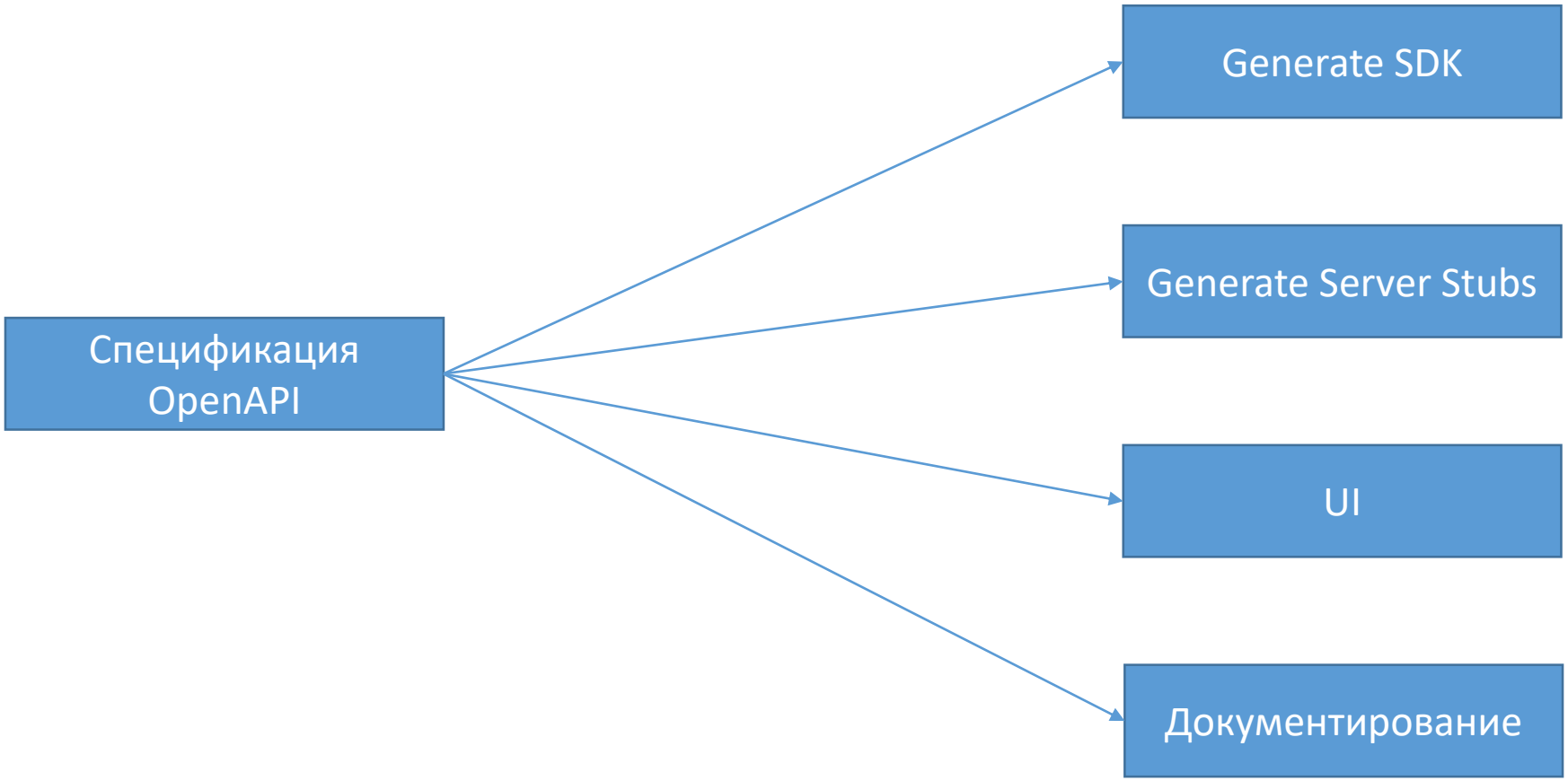
Как получить спецификацию?



Как получить спецификацию?



Что можно получить из спецификации?



Интеграция с веб-сервисом

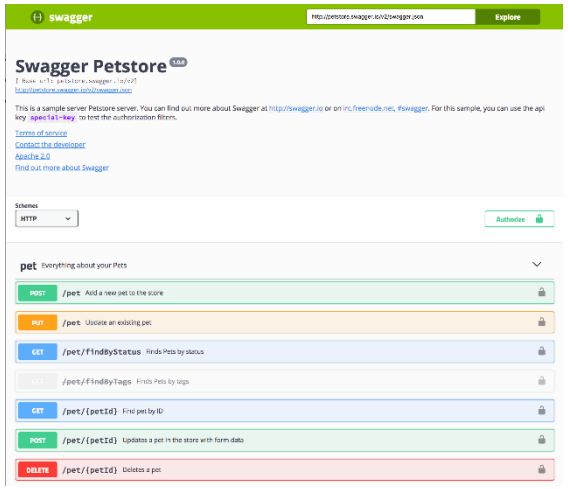
Backend
Server

Swagger

SwaggerUI

Спецификация OpenAPI

Web - приложение



Интеграция с веб-сервисом

Устанавливаем nuget-пакет:

Install-Package Swashbuckle

Автоматически создастся код инициализации

```
10  public class SwaggerConfig
11  {
12      public static void Register()
13      {
14          var thisAssembly = typeof(SwaggerConfig).Assembly;
15
16          GlobalConfiguration.Configuration
17              .EnableSwagger(c =>
18                  {
19                      c.SingleApiVersion("v1", "WebApplication1");
20                  })
21              .EnableSwaggerUi();
22
23      }
```

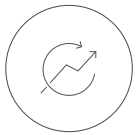
Наследование ViewModels

```
internal class PolymorphismSchemaFilter : BasePolymorphismFilter, ISchemaFilter
{
    private HashSet<Type> types;

    public void Apply(Schema schema, SchemaRegistry schemaRegistry, Type type)
    {
        // регистрируем базовый тип.
        if (!baseType.IsGenericType && !TypeMapping.ContainsKey(WebApiConfig.JsonTypeNameFormatter(baseType)))...

        var linkToParentSchema = new Schema...;

        if (!schemaRegistry.Definitions.ContainsKey(WebApiConfig.JsonTypeNameFormatter(baseType)))...

        var parentSchema = schemaRegistry.Definitions[WebApiConfig.JsonTypeNameFormatter(baseType)];

        schema.allOf = new List<Schema>
        {
            linkToParentSchema,
        };

        var properties = type.GetProperties(BindingFlags.DeclaredOnly | BindingFlags.Instance | BindingFlags.Public);

        // оставим только свойства текущего класса
        schema.properties = schema.properties.Where(pair => properties.Any(p => p.Name.ToLowerInvariant() == pair.Key.ToLowerInvariant())).ToDictionary(pair => pair.Key, pair =>
    }
}
```

Выбор инструмента кодогенерации

Swagger Codegen

Генерация сервер Stub на многих языках



Постоянное развитие

Генерация готовых SDK

NSwag

Генерация C# и TypeScript client/proxy

Различные способы вызова генерации: command-line, msbuild, in code via nuget, GUI

Шаблоны liquid для генерации

Кодогенерация

```
{% if HasDescription -%}
/// <summary>{{ Description | csharpdocs }}</summary>
{% endif -%}
{% if ExtensionData["x-loymax-type"] -%}
[JsonExtensionData("x-loymax-type", "{{ExtensionData["x-loymax-type"]}}")]
{% endif -%}
{% if HasDiscriminator -%}
[Newtonsoft.Json.JsonConverter(typeof(JsonInheritanceConverter), "{{ Discriminator }}")]
{% for derivedClass in DerivedClasses -%}
[JsonInheritanceAttribute("{{ derivedClass.Discriminator }}", typeof({{ derivedClass.ClassName }}))]
{% endfor -%}
{% endif -%}
[System.CodeDom.Compiler.GeneratedCode("NJsonSchema", "{{ ToolchainVersion }}")]
{% if InheritsExceptionSchema -%}
[Newtonsoft.Json.JsonObjectAttribute]
```

```
/// <summary>Модель значения атрибута рекламного материала.</summary>
[JsonExtensionData("x-loymax-type", "Loymax.Business.Announcement.Model.AttributeValueModelBase, Loymax.Business.Announcement.Model")]
[System.CodeDom.Compiler.GeneratedCode("NJsonSchema", "9.10.72.0 (Newtonsoft.Json v11.0.0.0)")]
public partial class AttributeValueModelBase
{
    /// <summary>Id атрибута.</summary>
    [Newtonsoft.Json.JsonProperty("attributeId", Required = Newtonsoft.Json.Required.Default, NullValueHandling = Newtonsoft.Json.Null
    public int? AttributeId { get; set; }

    /// <summary>Логическое имя атрибута.</summary>
    [Newtonsoft.Json.JsonProperty("attributeLogicalName", Required = Newtonsoft.Json.Required.Default, NullValueHandling = Newtonsoft.
    public string AttributeLogicalName { get; set; }

    /// <summary>Имя атрибута.</summary>
    [Newtonsoft.Json.JsonProperty("attributeName", Required = Newtonsoft.Json.Required.Default, NullValueHandling = Newtonsoft.Json.Nu
    public string AttributeName { get; set; }
```

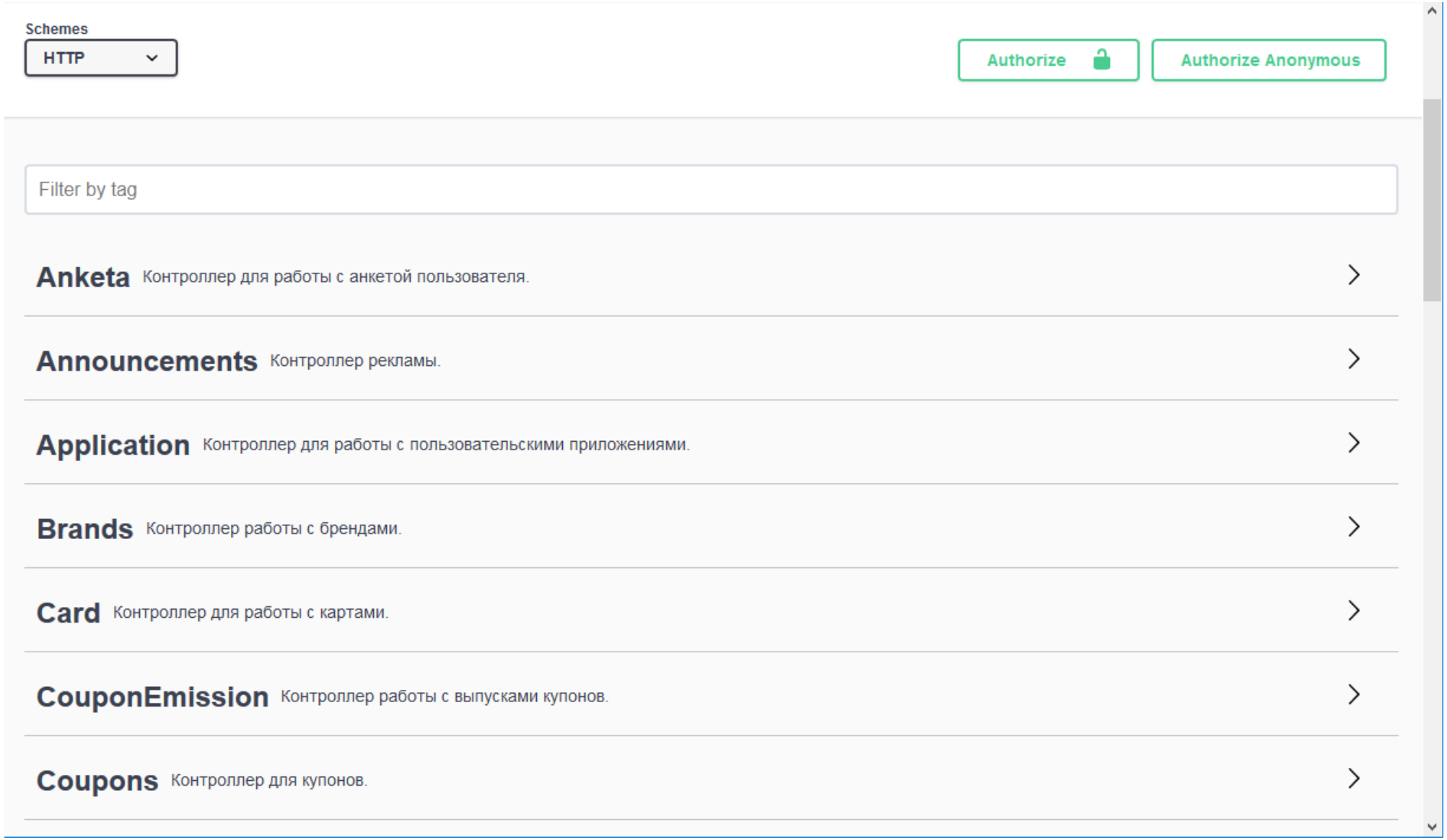
Адаптация SwaggerUI

InjectStylesheets

InjectJavascript

```
/// <summary>Конфигурирование настроек SwaggerUI.</summary>
/// <param name="config"><see cref="HttpConfiguration"/>.</param>
/// <param name="swaggerUiConfig"><see cref="SwaggerUiConfig"/>.</param>
protected virtual void OnEnableSwaggerUi(HttpConfiguration config, SwaggerUiConfig swaggerUiConfig)
{
    swaggerUiConfig.EnableDiscoveryUrlSelector();
    var assembly = typeof(StartupComponent).Assembly;
    swaggerUiConfig.UImaxDisplayedTags(200);
    swaggerUiConfig.ShowExtensions(true);
    swaggerUiConfig.CustomAsset("ext/jquery-3-3-1-min-js", assembly, "Loymax.Common.WebApi.SwaggerExtension.jquery-3.3.1.min.js");
    swaggerUiConfig.CustomAsset("ext/auth-js", assembly, "Loymax.Common.WebApi.SwaggerExtension.auth.js");
    swaggerUiConfig.InjectJavaScript(assembly, "Loymax.Common.WebApi.SwaggerExtension.sync-loader.js");
}
```

Адаптация SwaggerUI



Загрузка файлов из SwaggerUI

```
public class AddFileUploadFilter : IOperationFilter
{
    public void Apply(Operation operation, SchemaRegistry schemaRegistry, ApiDescription apiDescription)
    {
        var fileUploadAttribute = apiDescription.GetControllerAndActionAttributes<SwaggerAddFileUploadAttribute>();
        foreach (var attr in fileUploadAttribute)
        {
            operation.consumes.Add("multipart/form-data");
            if (operation.parameters == null)
            {
                operation.parameters = new List<Parameter>();
            }

            operation.parameters.Add(new Parameter { name = "file", @in = "formData", description = "file to upload", required = attr.Required, type = "file" });
        }
    }
}
```

Выводы

1. Более 30 наших клиентов интегрировались с нами используя SDK.
2. 3 года мы используем эту технологию – как результат, довольные тестировщики и разработчики.
3. Технология открыта и расширяема, поэтому должна подойти и вам.

Спасибо за внимание