



# Версионирование API

Доронкин Даниил  
[www.rubius.com](http://www.rubius.com)

# Что обсудим

- Версионирование;
- Подходы к управлению версиями;
- SemVer;
- AspNet-api-versioning;
- Swagger интеграция.

# Ассоциации

- Уникальное имя;
- Эволюция ПО;
- Стадия готовности;
- Минимальное документирование.

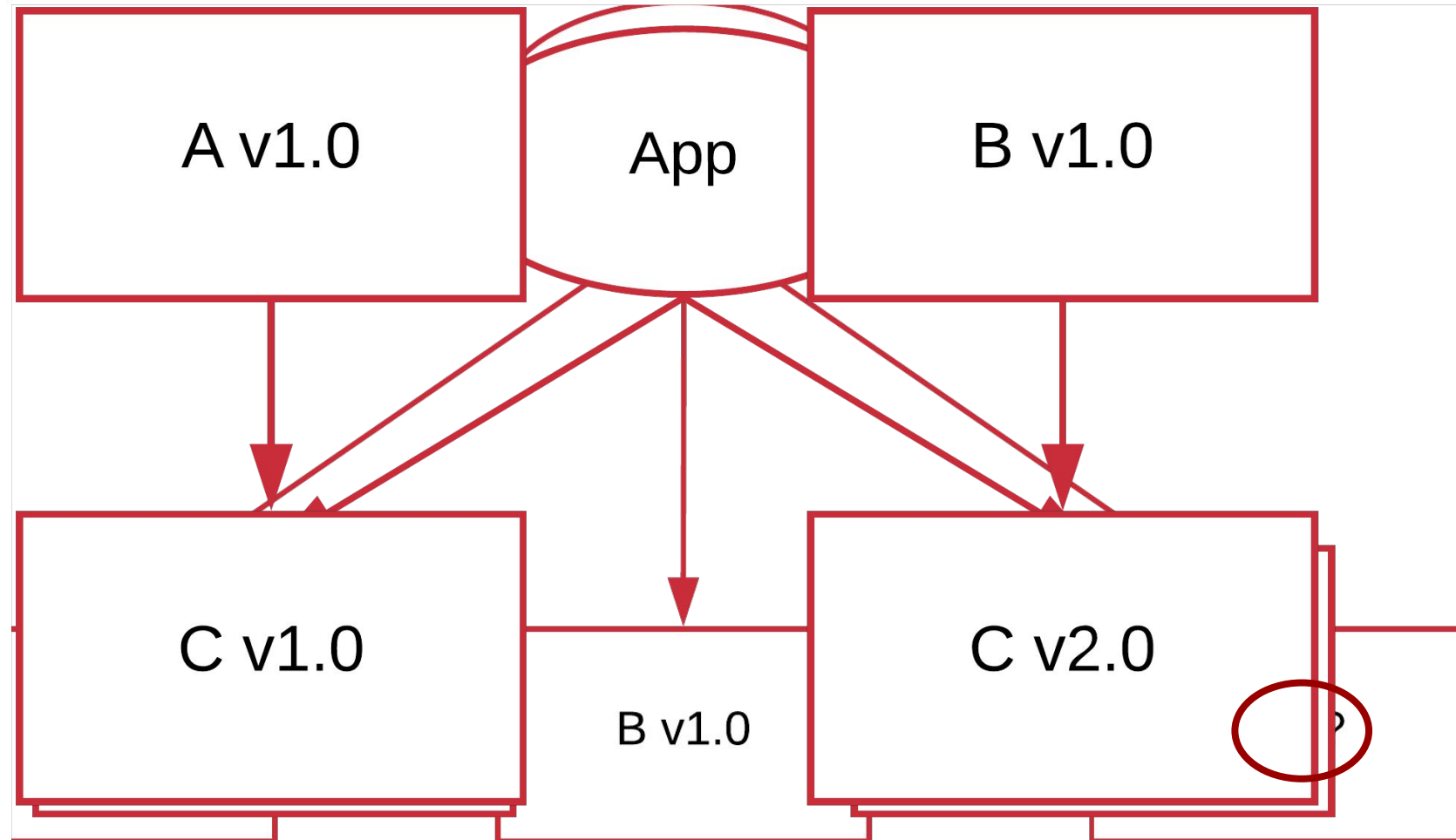
# Примеры

- C# **7.2**, Python **3.6.4**;
- .net core **2.0.6**, NodeJs **9.7.1**;
- VueJs **2.0**, Swashbuckle **2.3.0**;
- <http://sdb.amazonaws.com?...&Version=2009-04-15&...> ;
- Corel Draw **11**;
- Windows **2000**;
- Delphi **XE**;
- ...

# Зачем управлять версиями

- Контроль изменений и жизненного цикла;
- Прозрачная работы с зависимостями.

# Dependency hell



# Demo

DependencyHellNet - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test ReSharper Analyze Window Help

Quick Launch (Ctrl+Q)

Doronkin Daniil

Debug Any CPU DependencyHellNet


DependencyHellNet

NuGet - Solution


Browse Installed Updates 2 Consolidate 2

Search (Ctrl+E) Include prerelease

Package source: nuget.org

**Newtonsoft.Json** by James Newton-King v11.0.2


Json.NET is a popular high-performance JSON framework for .NET

**AutoMapper** by Jimmy Bogard v6.2.2

A convention-based object-object mapper. AutoMapper uses a fluent configuration API to define an object-object mapping strategy. AutoMapper uses a convention-based matching algorithm to match up...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

☐ Do not show this again

**AutoMapper**

Version(s) - 2

	Project	Version
<input type="checkbox"/>	DependencyHellNet	
<input type="checkbox"/>	dto	
<input checked="" type="checkbox"/>	moduleA	4.2.1
<input type="checkbox"/>	ModuleAConsole	
<input checked="" type="checkbox"/>	moduleB	6.2.2
<input type="checkbox"/>	ModuleBConsole	

Installed: multiple versions installed Uninstall

Version: Latest stable 6.2.2 Install

Find Results Web Publish Activity Package Manager Console Error List ... Breakpoints Output

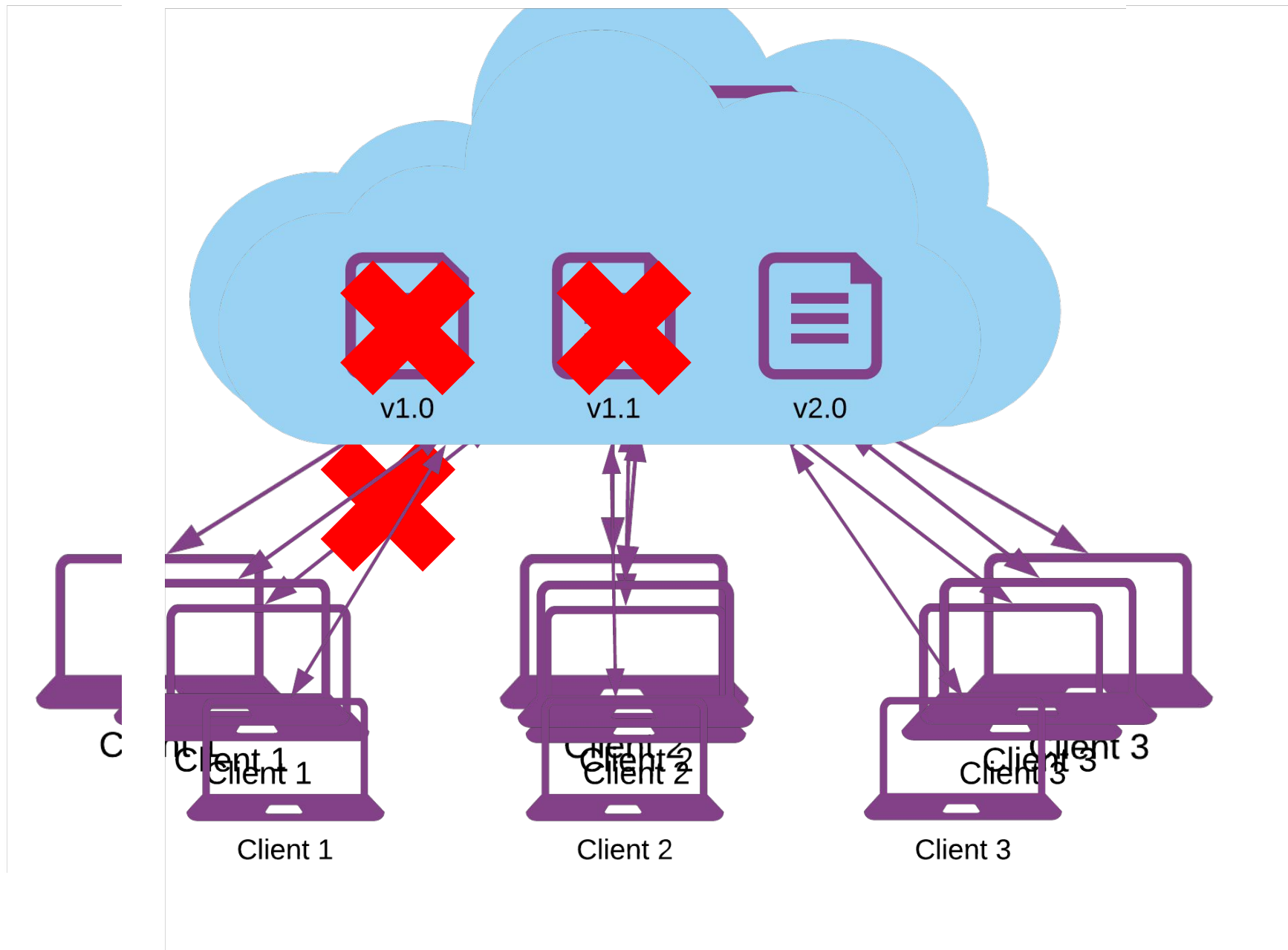
Ready

Add to Source Control



# Инструменты

- Nuget;
- Npm;
- Gb.
- ...



# Существующие подходы

- Целое число: Corel Draw 10;
- Текст: Delphi XE;
- Семантическое версионирование: .net core 2.0.6;
- Стадия разработки: ASP.NET SignalR Core 1.0.0-alpha1;
- Календарное версионирование: Rider 2018.1

...

# SemVer

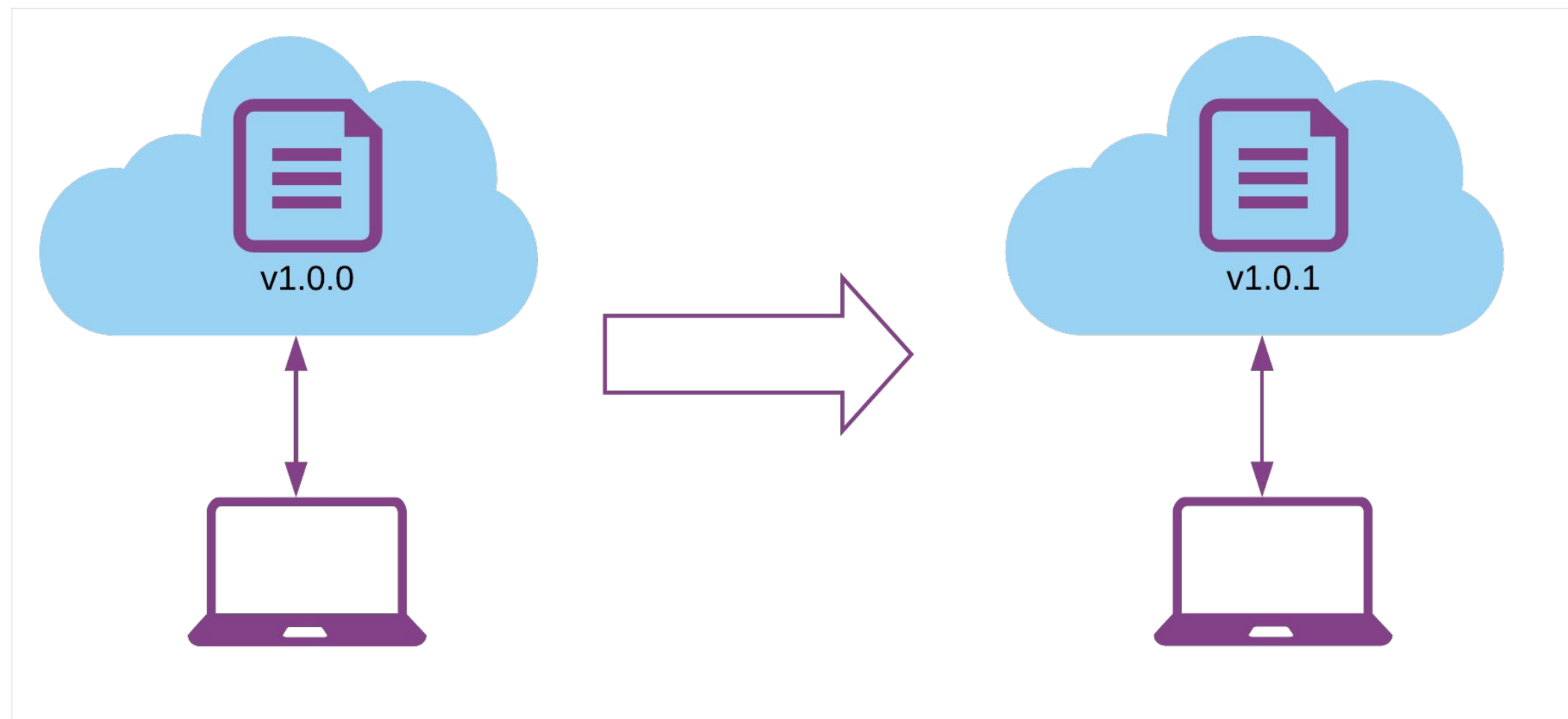
- X.Y.Z, где X, Y и Z – неотрицательные целые числа;
- X – мажорная версия, Y – минорная версия, Z – патч-версия;

1. 5. 4

major minor patch

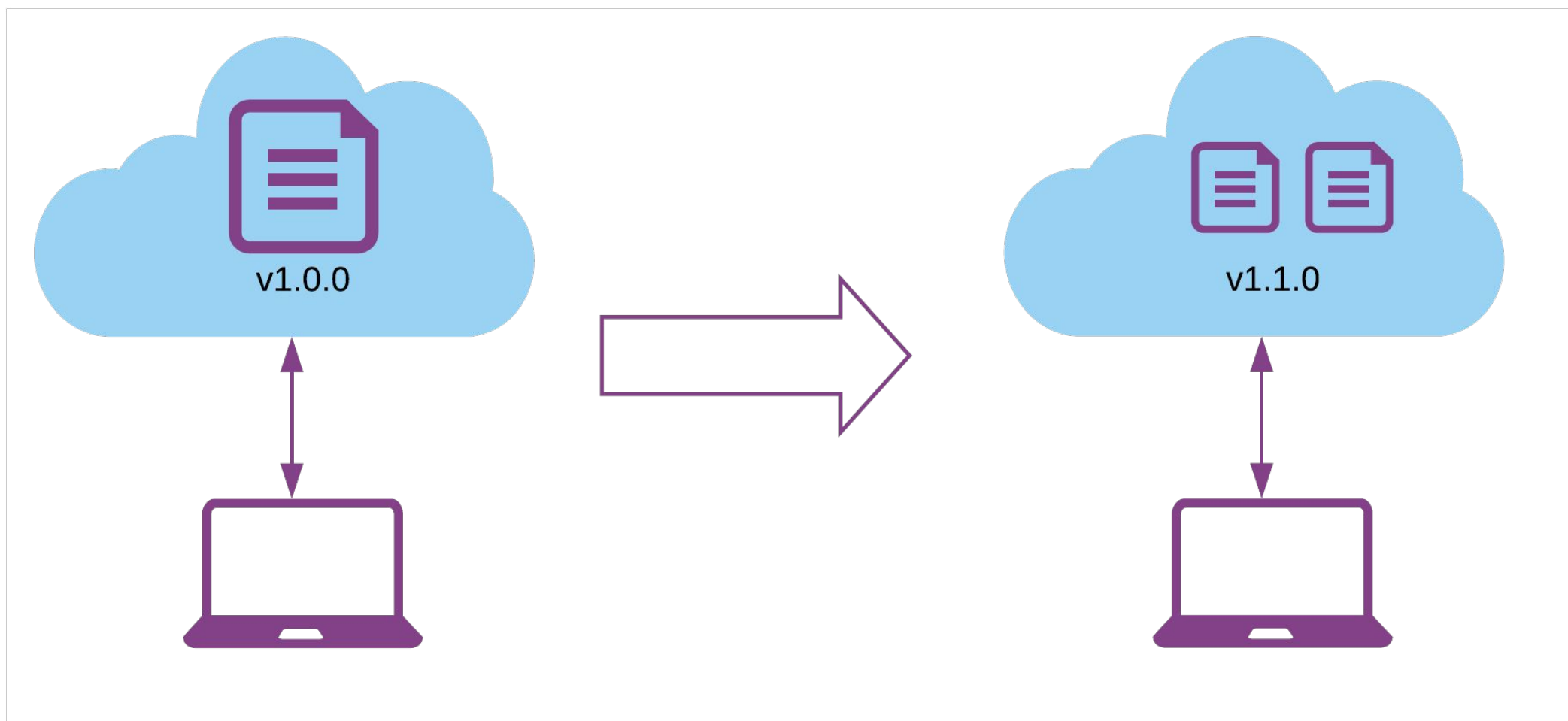
# Patch версия

- Инкрементируем после баг-фиксов.



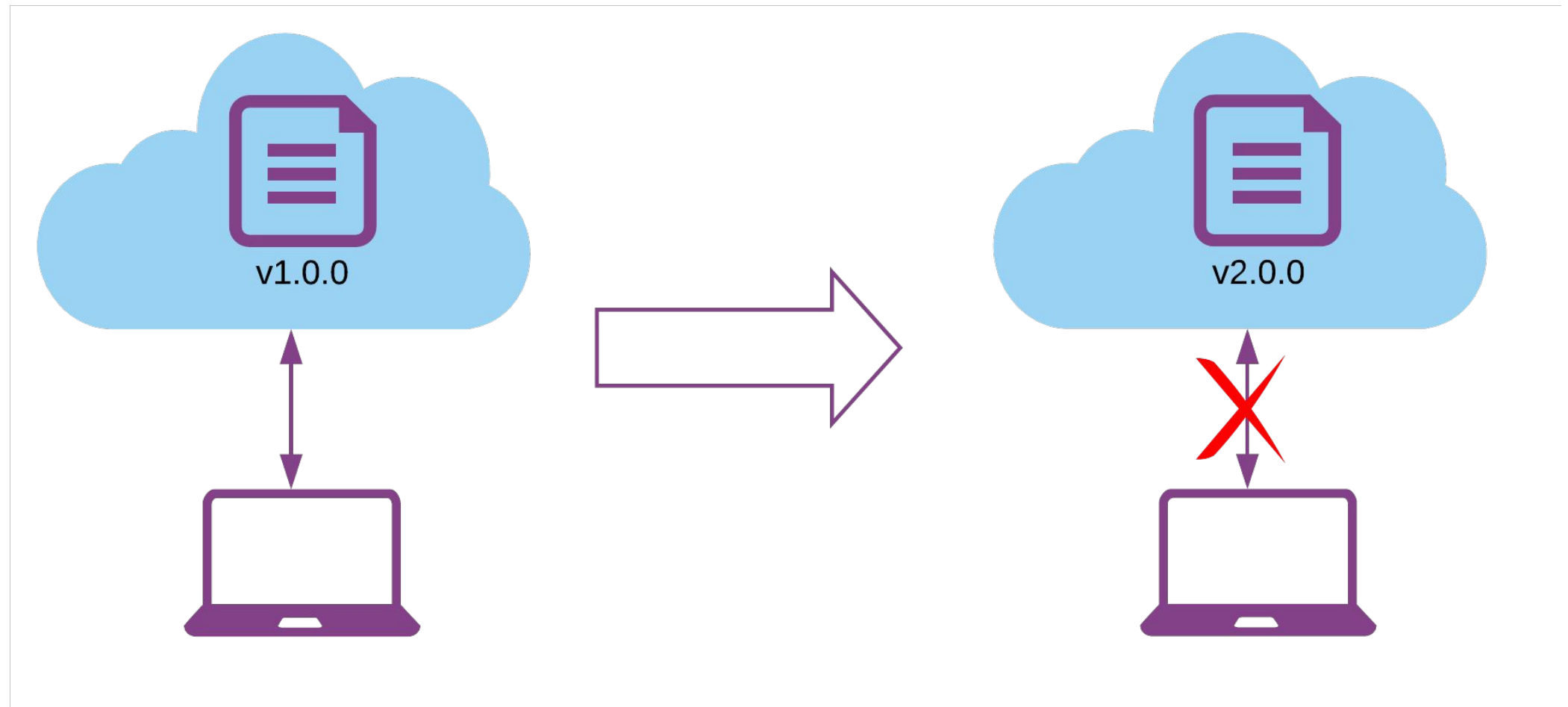
# Minor версия

- Добавление совместимой функциональности.



# Major версия

- Breaking changes.



# Расширение номера версии

- Суффикс для отображения стадии разработки;  
**1.0.0-beta2**
- Сборочные метаданные.  
**1.0.0-beta+exp.sha.5114f85**



# Изменения версии

- Версия O.Y.Z до первого релиза;
- После релиза содержимое пакета не изменяется.

# Production?

- Да – укажите начальную версию 1.0.0;
- Нет – начните с версии 0.1.0.

# Функциональность устарела

- Увеличьте `minor`-версию: 1.0.0 → 1.1.0;
- Увеличьте `major`-версию после удаления: 1.1.0 → 2.0.0.

# Стадия разработки

- Укажите идентификатор в названии версии: 2.0.0-alpha;
- 2.0.0-alpha < 2.0.0-beta < 2.0.0 < 3.0.0-alpha < 3.0.0.

# Размер номера версии

- Без ограничений.

\* чем короче – тем лучше :)

# Внутренние зависимости обновлены

- Могут быть рассмотрены как внутренние изменения;
- Обновлены для правки багов – patch-релиз.

## Breaking changes в minor релизе

- Верните обратную совместимость в новом minor релизе;
- Отрадите данную проблему в документации;
- Проинформируйте пользователей.

# Breaking changes вне спецификации

pet Everything about your Pets		Find out more: <a href="http://swagger.io">http://swagger.io</a>
POST	/pet Add a new pet to the store	🔒
PUT	/pet Update an existing pet	🔒
GET	/pet/findByStatus Finds Pets by status	🔒
GET	<del>/pet/findByTags</del> Finds Pets by tags	🔒
GET	/pet/{petId} Find pet by ID	🔒
POST	/pet/{petId} Updates a pet in the store with form data	🔒
DELETE	/pet/{petId} Deletes a pet	🔒
POST	/pet/{petId}/uploadImage uploads an image	🔒



# Breaking changes вне спецификации

- Ориентируйтесь по ситуации;
- Попробуйте решить проблему напрямую с клиентом;
- Обновите спецификацию в minor релизе и следом опубликуйте major релиз с исправлениями.

# SemVer 1.0.0 – 2.0.0

+ Сборочные метаданные;

1.0.0+20130313144700

1.0.0-beta+exp.sha.5114f85

– SemVerTag.

vX.Y.Z

# Преимущества

- Польза от номера версии;
- Формализованный набор правил;
- Простота.

# GitVersion



SemVer по мотивам git истории.

# CalVer

- Упрощенная семантика;
- Чувствительность к дате.

2018.1

year

release

# Microsoft aspnet-api-versioning

- Управление версиями API;
- Open source;
- Open closed принцип;

# Подключение

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddMvcCore().AddVersionedApiExplorer(
        options => { options.SubstituteApiVersionInUrl = true; });

    services.AddMvc();

    services.AddApiVersioning(options => { options.ReportApiVersions = true; });
}
```

# Конфигурация через атрибуты

```
[ApiVersion("1.0")]
[ApiVersion("2.0")]
[Route("api/v{version:apiVersion}/{controller}")] //url path
[Route("api/{controller}")] //query string
public class AttributesBasedController : Controller
{
    [MapToApiVersion("1.0")]
    [HttpGet]
    public string Get() => "V1";

    [MapToApiVersion("2.0")]
    [HttpPost]
    public string Post() => "V1";
}
```



# Передача версии в запросе

## Request URL

```
http://localhost:5000/api/v1/attributesBased
```

## Request URL

```
http://localhost:5000/api/AttributesBased?api-version=1.0
```

## Response headers

```
api-supported-versions: 1.0, 2.0, 3.0  
content-type: text/plain; charset=utf-8  
date: Sun, 08 Apr 2018 21:21:49 GMT  
server: Kestrel  
transfer-encoding: chunked
```

# Стратегии передачи версии

- URL;
- Content negotiation;
- Query parameter.

# URL

- `https://app.com/api/v1.0/<params>`
- Отдельная точка доступа под каждую версию;
- Не REST;
- + Версия всегда на виду;
- + Легко тестировать.

# Content negotiation

- accept, content-type;
- Тестировать сложнее;
- Требуется обработка заголовков;
- + Один ресурс – одна точка доступа;
- + REST;

# Query parameter

- `https://app.com/api/<params>?api-version=1.0`
- Размер URL;
- Непрозрачное поведение если Query parameter не передан.
- + Одна и та же endpoint для всех версий;
- + Версия видна в адресной строке;

# Options конфигурация

```
services.AddApiVersioning( options =>
{
    options.ReportApiVersions = true;
    options.Conventions.Controller<ConventionsBasedController>()
        .HasApiVersion(apiV1)
        .HasApiVersion(apiV2)
        .Action(c => c.Get()).MapToApiVersion(apiV1)
        .Action(c => c.Post()).MapToApiVersion(apiV1)
        .Action(c => c.Put()).MapToApiVersion(apiV1)
        .Action(c => c.Delete()).MapToApiVersion(apiV1)
        .Action(c => c.GetV2()).MapToApiVersion(apiV2)
        .Action(c => c.PostV2()).MapToApiVersion(apiV2)
        .Action(c => c.PutV2()).MapToApiVersion(apiV2)
        .Action(c => c.DeleteV2()).MapToApiVersion(apiV2);
});
```



# Устаревшая версия

```
services.AddApiVersioning( options =>
{
    options.Conventions.Controller<ConventionsBasedController>()
        .HasDeprecatedApiVersion(apiV1);
});

[ApiVersion("1.0", Deprecated = true)]
[ApiVersion("2.0")]
[ApiVersion("3.0")]
[Route( "api/v{version:apiVersion}/[controller]" )]
[Route( "api/[controller]" )]
public class AttributesBasedController : Controller
```

## Response headers

```
api-deprecated-versions: 1.0
api-supported-versions: 2.0, 3.0
```

# Версия по умолчанию

```
services.AddApiVersioning( options =>
{
    options.ReportApiVersions = true;
    options.AssumeDefaultVersionWhenUnspecified = true;
    options.DefaultApiVersion = new ApiVersion(new DateTime(2017,1,1));
});
```

## Request URL

```
http://localhost:5000/api/NotSpecified?api-version=2017-01-01
```

## Response headers

```
api-supported-versions: 2017-01-01
content-type: text/plain; charset=utf-8
date: Mon, 09 Apr 2018 20:11:36 GMT
```



# Out of version

```
[ApiVersionNeutral]  
[Route( "api/[controller]" )]  
public class HealthController : Controller
```

## Request URL

```
http://localhost:5000/api/Health/ping
```

## Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 09 Apr 2018 20:21:24 GMT  
server: Kestrel  
transfer-encoding: chunked
```

# Кастомизация

- Способ отправки версии в запросе;
- Кастомные version-атрибуты;
- Переопределение Error response;
- ...

# Swagger интеграция

```
services.AddSwaggerGen(  
    options =>  
    {  
        var provider = services  
            .BuildServiceProvider()  
            .GetRequiredService<IApiVersionDescriptionProvider>();  
  
        foreach (var description in provider.ApiVersionDescriptions)  
        {  
            options.SwaggerDoc(  
                description.GroupName,  
                CreateInfoForApiVersion(description));  
        }  
  
        options.OperationFilter<SwaggerDefaultValues>();  
    });
```

# Swagger интеграция

```
app.UseSwagger();
app.UseSwaggerUI(
    options =>
    {
        // build a swagger endpoint for each discovered API version
        foreach (var description in provider.ApiVersionDescriptions)
        {
            options.SwaggerEndpoint(
                $"/swagger/{description.GroupName}/swagger.json",
                description.GroupName.ToUpperInvariant());
        }
    });
```

# Demo

# Подводные камни

- Media type API version reader;
- Microsoft.AspNetCore.OData;
- Issues на Github.

# Преимущества

- Простой и удобный API.

# Преимущества

- Богатая функциональность;
- Кастомизация.



# Преимущества

- Безболезненное внедрение.

# Преимущества

- Swagger интеграция.

# ИСТОЧНИКИ

- <https://github.com/Microsoft/aspnet-api-versioning/wiki;>
- [https://semver.org/;](https://semver.org/)
- [https://gitversion.readthedocs.io/en/latest/;](https://gitversion.readthedocs.io/en/latest/)
- [https://github.com/ddydeveloper/ApiVersioning;](https://github.com/ddydeveloper/ApiVersioning)
- [https://github.com/ddydeveloper/DependencyHellExample;](https://github.com/ddydeveloper/DependencyHellExample)
- [https://www.youtube.com/watch?v=ngcigr\\_8oxw&t=109s.](https://www.youtube.com/watch?v=ngcigr_8oxw&t=109s)



Спасибо за внимание

[www.rubius.com](http://www.rubius.com)  
[ddydeveloper@gmail.com](mailto:ddydeveloper@gmail.com)