技术交流

微信：

luckylucky421302

加微信备注k8s即可进群交流学习



在[API conventions doc](https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md" \t "_blank)中描述了API的全部协议。

在[API Reference](https://kubernetes.io/docs/reference)文档中描述了API的端点、资源类型和示例。

在[Controlling API Access doc](https://kubernetes.io/docs/reference/access-authn-authz/controlling-access/)中讨论了通过远程访问API的相关问题。

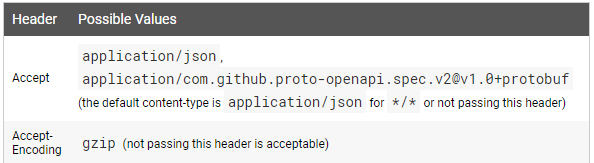
Kubernetes API是系统声明式配置架构的基础。Kubectl命令行工具被用于创建、更新、删除、获取API对象。Kubernetes通过API资源存储自己的序列化状态(这些状态存储在etcd中)。Kubernetes被分成多个组件，各组件通过API相互交互。

**API变更**

根据经验，任何成功的系统都需要随着新的用例出现或现有用例发生变化的情况下，进行相应的进化与调整。因此，我们希望Kubernetes API也可以保持持续的进化和调整。同时，在较长一段时间内，我们也希望与现有客户端版本保持良好的向下兼容性。一般情况下，增加新的API资源和资源字段不会导致向下兼容性问题发生；但如果是需要删除一个已有的资源或者字段，那么必须通过[API废弃流程](https://kubernetes.io/docs/reference/deprecation-policy/)来进行。可参考[API变更文档](https://git.k8s.io/community/contributors/devel/sig-architecture/api_changes.md)，了解兼容性变更的要素以及如何变更API的流程

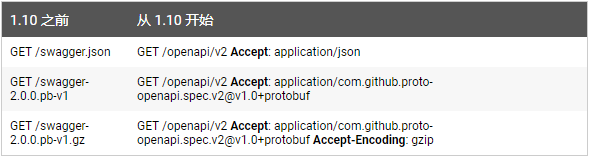
**OpenAPI和API Swagger 定义**

完整的API详细文档使用 [OpenAPI](https://www.openapis.org/" \t "_blank)生成，随着Kubernetes 1.10 版本的正式启用，Kubernetes API服务通过/openapi/v2接口提供OpenAPI规范。通过设置 HTTP 标头规定了请求的结构。



在1.14版本之前，区分结构的接口通过(/swagger.json, /swagger-2.0.0.json, /swagger-2.0.0.pb-v1, /swagger-2.0.0.pb-v1.gz) 提供不同格式的 OpenAPI规范。但是这些接口已经被废弃，并且已经在 Kubernetes 1.14 中被删除。

**获取OpenAPI 规范的例子**:



Kubernetes实现了另一种基于Protobuf的序列化格式，该格式主要用于集群内通信，并在[设计方案](https://github.com/kubernetes/community/blob/master/contributors/design-proposals/api-machinery/protobuf.md" \t "_blank)中进行了说明，每个模式的IDL文件位于定义API对象的Go软件包中。 在 1.14 版本之前， Kubernetes apiserver 也提供API服务用于返回[Swagger v1.2](http://swagger.io/" \t "_blank)，Kubernetes API规范通过/swaggerapi接口，但是这个接口已经被废弃，并且在 Kubernetes 1.14 中已经被移除。

**API 版本**

为了使删除字段或者重构资源更加容易，Kubernetes支持多个API版本。每一个版本都在不同API路径下，例如/api/v1或者/apis/extensions/v1beta1。我们选择在API级别进行版本化，而不是在资源或字段级别进行版本化，以确保API提供清晰、一致的系统资源和行为视图，并控制对已废止的API和/或实验性API的访问。 JSON和Protobuf序列化模式遵循架构更改的相同准则 - 下面的所有描述都同时适用于这两种格式。请注意，API版本控制和软件版本控制只有间接相关性。[API和发行版本建议](https://git.k8s.io/community/contributors/design-proposals/release/versioning.md" \t "_blank)描述了API版本与软件版本之间的关系。不同的API版本名称意味着不同级别的软件稳定性和支持程度。每个级别的标准在[API变更文档](https://git.k8s.io/community/contributors/devel/sig-architecture/api_changes.md" \l "alpha-beta-and-stable-versions" \t "_blank)中有更详细的描述。 内容主要概括如下：

**alpha-内部测试版本：**

1）版本名称包含了alpha (例如：v1alpha1)。

2）可能是有缺陷的。启用该功能可能会带来隐含的问题，默认情况是关闭的。

3）支持的功能可能在没有通知的情况下随时删除。

4）API的更改可能会带来兼容性问题，但是在后续的软件发布中不会有任何通知。

5）由于bugs风险的增加和缺乏长期的支持，推荐在短暂的集群测试中使用。

**beta-公测版本：**

1）版本名称包含了beta(例如:v2beta3)。

2）代码已经测试过。启用该功能被认为是安全的，功能默认已启用。

3）所有已支持的功能不会被删除，细节可能会发生变化。

4）对象的模式和/或语义可能会在后续的beta测试版或稳定版中以不兼容的方式进行更改。 发生这种情况时，我们将提供迁移到下一个版本的说明。 这可能需要删除、编辑和重新创建API对象。执行编辑操作时需要谨慎行事，这可能需要停用依赖该功能的应用程序。

5）建议仅用于非业务关键型用途，因为后续版本中可能存在不兼容的更改。 如果有多个可以独立升级的集群，则可以放宽此限制。

**stable-稳定版本：**

1）版本名称是vX，其中X是整数。

2）功能的稳定版本将出现在许多后续版本的发行软件中。

**API组**

为了更容易地扩展Kubernetes API，我们实现了*[API组](https://git.k8s.io/community/contributors/design-proposals/api-machinery/api-group.md" \t "_blank)*。 API组在REST路径和序列化对象的**apiVersion**字段中指定。目前有几个API组正在使用中：

1. 核心组（通常被称为遗留组）位于REST路径/api/v1并使用apiVersion：v1。

2）指定的组位于REST路径/apis/$GROUP\_NAME/$VERSION，并使用apiVersion：$GROUP\_NAME/$VERSION（例如apiVersion：batch/v1）。在[Kubernetes API参考](https://kubernetes.io/docs/reference/)中可以看到支持的API组的完整列表。

社区支持使用以下两种方式来提供自定义资源对API进行扩展[自定义资源](https://kubernetes.io/docs/concepts/api-extension/custom-resources/)：

1）[CustomResourceDefinition](https://kubernetes.io/docs/tasks/access-kubernetes-api/extend-api-custom-resource-definitions/)适用于具有非常基本的CRUD需求的用户。

2）需要全套Kubernetes API语义的用户可以实现自己的apiserver，并使用[聚合器](https://kubernetes.io/docs/tasks/access-kubernetes-api/configure-aggregation-layer/)为客户提供无缝的服务。